



HAL
open science

Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages

Paul Oumaziz

► **To cite this version:**

Paul Oumaziz. Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages. Mécanique des solides [physics.class-ph]. Université Paris Saclay (COMUE), 2017. Français. NNT : 2017SACLN030 . tel-01588293

HAL Id: tel-01588293

<https://theses.hal.science/tel-01588293>

Submitted on 15 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2017SACLN030

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PARIS-SACLAY
PRÉPARÉE À L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Ecole doctorale n°579
Sciences mécaniques et énergétiques, matériaux et géosciences
Spécialité de doctorat : Solides, structures et matériaux

par

M. PAUL OUMAZIZ

Une méthode de décomposition de domaine mixte non-intrusive pour le
calcul parallèle d'assemblages

Thèse présentée et soutenue à l'École Normale Supérieure de Cachan, le 7 juillet 2017.

Composition du Jury :

M. FRÉDÉRIC MAGOULES	Professeur École CentraleSupélec	(Président)
M. ALBERTO CORIGLIANO	Professeur Politecnico di Milano	(Rapporteur)
M. DAVID DUREISSEIX	Professeur INSA Lyon	(Rapporteur)
M. MICKAËL ABBAS	Ingénieur-chercheur EDF-R&D	(Examineur)
M. STÉPHANE GUINARD	Ingénieur-chercheur Airbus	(Examineur)
M. JEAN-CHARLES PASSIEUX	Maître de conférence (HDR) Institut Clément Ader - Université de Toulouse	(Examineur)
M. PIERRE-ALAIN BOUCARD	Professeur École Normale Supérieure Paris-Saclay	(Directeur de thèse)
M. PIERRE GOSSELET	Chercheur CNRS (HDR) École Normale Supérieure Paris-Saclay	(Co-directeur de thèse)

Titre : Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages

Keywords : Décomposition de domaine, Calcul non-intrusif, Assemblages, Contact

Résumé : Les assemblages sont des éléments critiques pour les structures industrielles. De fortes non-linéarités de type contact frottant, ainsi que des précharges mal maîtrisées rendent complexe tout dimensionnement précis. Présents en très grand nombre sur les structures industrielles (quelques millions pour un A380), cela implique de raffiner les modèles localement et donc de gérer des problèmes numériques de très grandes tailles. Les nombreuses interfaces de contact frottant sont des sources de difficultés de convergence pour les simulations numériques. Il est donc nécessaire de faire appel à des méthodes robustes. Il s'agit d'utiliser des méthodes itératives de décomposition de domaine, permettant de gérer des modèles numériques extrêmement grands, couplées à des techniques adaptées afin de prendre en compte les non-linéarités de contact aux interfaces entre sous-domaines. Ces méthodes de décomposition de domaine restent encore très peu utilisées dans un cadre industriel. Des développements internes aux codes éléments finis sont souvent nécessaires et freinent ce transfert du monde académique au monde industriel.

Nous proposons, dans ces travaux de thèse, une mise-en-oeuvre non intrusive de ces méthodes de décomposition de domaine : c'est-à-dire sans développement au sein du code source. En particulier, nous nous intéressons à la méthode Latin dont la philosophie est particulièrement adaptée aux problèmes non linéaires. La structure est décomposée en sous-domaines reliés entre eux au travers d'interfaces. Avec la méthode Latin, les non-linéarités sont résolues séparément des aspects linéaires. La résolution est basée sur un schéma itératif à deux directions de recherche qui font dialoguer les problèmes linéaires globaux et les problèmes locaux non linéaires.

Au cours de ces années de thèse, nous avons développé un outil totalement non intrusif sous *Code_Aster* permettant de résoudre par une technique de décomposition de domaine mixte des problèmes d'assemblage. Les difficultés posées par le caractère mixte de la méthode Latin sont résolues par l'introduction d'une direction de recherche non locale. Des conditions de Robin sur les interfaces des sous-domaines sont alors prises en compte simplement sans modifier les sources de *Code_Aster*. Nous avons proposé une réécriture algébrique de l'approche multi-échelle assurant l'extensibilité de la méthode. Nous nous sommes aussi intéressés à coupler la méthode Latin en décomposition de domaine à un algorithme de Krylov. Appliqué uniquement à un problème sous-structuré avec interfaces parfaites, ce couplage permet d'accélérer la convergence. Des structures préchargées avec de nombreuses interfaces de contact frottant ont été traitées. Des simulations qui n'auraient pu être menées par un calcul direct sous *Code_Aster* ont été réalisées via cette stratégie de décomposition de domaine non intrusive.

Title : A non-invasive mixed domain decomposition for the parallel computation of assemblies

Keywords : Domain decomposition, Non-invasive computation, Assembly, Contact

Abstract : Assemblies are critical elements for industrial structures. Strong non-linearities such as frictional contact, as well as poorly controlled preloads make complex all accurate sizing. Present in large numbers on industrial structures (a few million for an A380), this involves managing numerical problems of very large size. The numerous interfaces of frictional contact are sources of difficulties of convergence for the numerical simulations. It is therefore necessary to use robust but also reliable methods. The use of iterative methods based on domain decomposition allows to manage extremely large numerical models. This needs to be coupled with adapted techniques in order to take into account the nonlinearities of contact at the interfaces between subdomains. These methods of domain decomposition are still scarcely used in industries. Internal developments in finite element codes are often necessary, and thus restrain this transfer from the academic world to the industrial world.

In this thesis, we propose a non-intrusive implementation of these methods of domain decomposition : that is, without development within the source code. In particular, we are interested in the Latin method whose philosophy is particularly adapted to nonlinear problems. It consists in decomposing the structure into sub-domains that are connected through interfaces. With the Latin method the non-linearities are solved separately from the linear differential aspects. Then the resolution is based on an iterative scheme with two search directions that make the global linear problems and the nonlinear local problems dialogue.

During this thesis, a totally non-intrusive tool was developed in *Code_Aster* to solve assembly problems by a mixed domain decomposition technique. The difficulties posed by the mixed aspect of the Latin method are solved by the introduction of a non-local search direction. Robin conditions on the subdomain interfaces are taken into account simply without modifying the sources of *Code_Aster*. We proposed an algebraic rewriting of the multi-scale approach ensuring the extensibility of the method. We were also interested in coupling the Latin method in domain decomposition to a Krylov algorithm. Applied only to a substructured problem with perfect interfaces, this coupling accelerates the convergence. Preloaded structures with numerous contact interfaces have been processed. Simulations that could not be carried out by a direct computation with *Code_Aster* were performed via this non-intrusive domain decomposition strategy.



A mon grand-père,

Remerciements

Je tiens à remercier premièrement mes deux directeurs Peter et PAB. Merci pour leur confiance durant ces trois belles années, merci pour leur réactivité et disponibilité qui m'ont toujours permis d'avancer efficacement. Merci Peter pour avoir toujours des idées plus ou moins farfelues, et ce même pendant la rédaction du manuscrit. Merci à PAB d'avoir calmé Peter de temps en temps, merci à tous les deux, faire une thèse avec vous c'était PARFAIT.

Merci à ceux qui m'ont aidé pour les quelques présoutenances : Stéphane, Werty, Benoît, Camille, Yoann, Paul-Baptiste, Charles, Snachou, Maxime, PAG, Maxence et Thomas.

Merci aussi aux permanents avec qui nous avons passé de bons moments, en particulier Manu, Yoann, Martin, Bruno.

Je souhaite aussi remercier Frisou et Philippe pour leur aide informatique. Merci à Philippe pour sa patiente et ses nombreuses compilations de code_aster, en particulier sur fusion.

Merci à Boubou, Rémi, Canis et Frisou pour la bonne ambiance qu'ils ont mise pendant ces années, dédicace particulière à Boubou pour la découverte des musiques balkaniques.

Un clin d'oeil à l'équipe de PSC pour les pauses sportives qui structurent la journée de travail : Werty, Silvère, Théau, Snachar... Je souhaite remercier l'ensemble des doctorants que j'ai pu côtoyer ; les collègues de 3^e année : Maxime, Camille, Basile, BVoillot, Rana, Marie-Pauline, Thomas, François et tant d'autres ; les plus jeunes : le Stagiaire, Snachou, Charles, le Douanier, le Corse, Marie, Maxence, notre fantastique Bhimal... ; les plus vieux : Wery, TTT, Python, Théau, Silvère, et ceux que j'oublie malheureusement. Si c'est aussi agréable de venir au labo, c'est en grande partie grâce à eux.

Si la thèse se passe bien et si un labo tourne bien c'est aussi grâce aux gestionnaires. Pour son efficacité, sa disponibilité et sa gentillesse je tiens à remercier chaleureusement Lydia qui a toujours été là pendant ces trois ans.

Je remercie l'équipe pédagogique du DGM avec qui j'ai pu travailler pendant mes missions d'enseignement et pour les quatre premières années de formation. Merci à Christine pour le suivi de ma scolarité au DGM et finalement la fin de la thèse depuis l'école doctorale. Un grand merci aussi à Danielle pour tous ces moments. Tu nous manques depuis ton départ pour UVSQ.

Je profite aussi de l'occasion pour remercier plus généralement tout ceux que j'ai rencontré pendant mes sept années à l'ENS, que ce soient mes collègues du BdE ou des autres associations et clubs, que ce soient mes collègues de promo et ceux des autres années, merci à eux, ces sept années sont passées à une vitesse folle.

Merci aussi aux personnes de l'administration et équipe de direction de l'ENS avec qui j'ai pu travailler pendant ma scolarité : M. Peyroche, M. Zalio, Mme Gobert, sans oublier ceux qui ont quitté l'ENS : Brigitte Vidal, Mme Verscheure ainsi que M. Savattero.

Je souhaite remercier ma famille qui, même sans comprendre exactement ce que je faisais, a toujours été là et le sera toujours par la suite. Merci pôpa, merci môman.

Enfin je souhaite remercier Rébecca avec qui je partage ma vie depuis presque 7 ans. Gros bisous ma chérie.

Table des matières

Introduction	1
1 État de l'art du traitement du contact en décomposition de domaine	7
1 Méthodes de Schwarz continue	9
1.1 Méthode de Schwarz alternée	10
1.2 Méthode de Schwarz parallèle	11
1.3 Méthode de Schwarz sans recouvrement	12
2 Méthodes de Schwarz discrètes	13
2.1 La Multiplicative Schwarz method	14
2.2 La Additive de Schwarz Method	15
2.3 La restricted additive Schwarz method	16
3 Approches primales et duales sans recouvrement	17
3.1 Généralités pour les problèmes sans recouvrements - notations	17
3.2 Approches primales	22
3.3 Approches duales	24
4 Généralités sur le traitement du contact frottant	28
4.1 Formulation du problème de contact frottant	28
4.2 Boucle interne : prise en compte des conditions limites de contact	30
4.3 Boucle externe : algorithme d'optimisation	33
5 Prise en compte du contact en décomposition de domaine	37
5.1 FETI pour le contact	37
5.2 Méthodes de Schwarz pour le contact	43
6 Bilan	43
2 Implémentation non-intrusive de la méthode Latin	46
1 Introduction - la méthode Latin : un solveur non-linéaire	47
1.1 Principes généraux de la méthode Latin	47
1.2 Quelques cas d'application	49
1.3 Méthode Latin multi-paramétrique	49
2 Méthode Latin appliquée à la décomposition de domaine mono-échelle	50

2.1	Obtention du problème	50
2.2	Convergence	52
2.3	Interprétation des directions de recherches	52
2.4	Etape linéaire intrusive	53
3	Vision non-intrusive par direction de recherche non-locale	54
3.1	Direction de recherche non-locale	54
3.2	Validation sur un cas simple 1D	59
3.3	Présentation d'une simulation 3D	62
4	Bilan	67
3	Gestion de l'étape linéaire : vision multi-échelle et accélération	69
1	Introduction	70
2	Approche multi-échelle pour un équilibre global	72
2.1	Définition d'une base macroscopique en déplacement	72
2.2	Reécriture de l'étape linéaire	75
2.3	Schéma de résolution du problème multi-échelle	80
3	Approche multi-échelle pour une continuité globale	86
3.1	Définition d'une base macroscopique en effort	86
3.2	Définition de l'étape linéaire	86
3.3	Schéma de résolution du problème en déplacement	88
4	Comparaison des deux approches - exemples numériques	89
4.1	Étude de l'extensibilité	89
4.2	Influence de la direction de recherche	94
4.3	Influence du paramètre de relaxation	96
5	Accélération de type Krylov	97
5.1	Principe de l'accélération de Krylov	97
5.2	Illustration	100
6	Bilan	103
4	Formulation quasi-statique pour les non-linéarités de contact	105
1	Méthode Latin dans un contexte quasi-statique	106
1.1	Mise en place du problème quasi-statique	107
1.2	Etape linéaire	108
1.3	Etape locale : interfaces parfaites	108
2	Etape locale : problème de contact aux interfaces	108
2.1	Formulation de l'étape locale pour un comportement de contact	108
2.2	Simplification de l'opérateur de direction de recherche non-locale	109
2.3	Résolution du problème de contact	111
2.4	Validation de l'implémentation des interfaces de contact	115
3	Assemblages pré-chargés	124
3.1	Objectif de la pré-charge	124
3.2	Ecriture via une interface de contact	124
3.3	Exemples numériques	126

4	Application à un cas 3D	128
5	Bilan	129
5	Implémentation numérique et exemples industriels	132
1	Introduction - principes et structures de <i>code_aster</i>	133
1.1	Principes de <i>code_aster</i> et du superviseur	133
1.2	Base de données JEVEUX - concept ASTER	134
2	Détails de l'implémentation et mise en donnée	135
2.1	Pré-traitement	135
2.2	Calcul Latin générique	138
3	Performance de la méthode pour des cas académiques	141
3.1	Etude du <i>speed-up</i>	141
3.2	Cas de contacts multiples	142
4	Quelques applications industrielles	148
4.1	Liaison de lanceur spatial	148
4.2	Cas test industriel	150
5	Bilan	156
	Conclusion	158
A	Précisions sur le traitement du contact	161
1	Démonstration	161
1.1	Cas du contact normal	161
1.2	Cas du frottement	163
B	Description des structures industrielles	165
1	Description des cas industriels	165
1.1	Bride	165
1.2	Liaison boulonnée Airbus	167
	Bibliographie	170

Introduction

Les structures industrielles sont dimensionnées selon de nombreux critères afin de respecter un cahier des charges précis. Dans le cadre de l'industrie aéronautique la sécurité des passagers et des équipages est l'élément le plus important à considérer lors de la conception d'un appareil. Cela passe en particulier par garantir l'intégrité physique de l'engin tout au long du cycle d'utilisation de l'appareil. Des critères de tenue mécanique des différents éléments constituant l'appareil sont définis afin de résister à des sollicitations à la fois variées mais aussi extrêmes. Par exemple les pièces constituant les réacteurs peuvent être soumises à des températures jusqu'à 1000 °C et doivent être dimensionnées pour résister à des impacts potentiels d'animaux. Les trains d'atterrissage doivent quant à eux supporter de façon brève et violente l'atterrissage et ce plus de 50 000 fois durant la vie de l'appareil. A ces grandes variétés de sollicitations viennent se greffer des comportements de fatigue à même de causer la ruine d'éléments vitaux pour la structure. Dimensionner de telles structures devient alors extrêmement complexe.

Plusieurs centaines de milliers de points de fixations sont dénombrés sur un avion allant même jusqu'à quelques millions pour un A380. Les éléments structuraux sont alors inévitablement affaiblis au niveau des zones d'assemblage. Les nombreux trous nécessaires à la mise en place des éléments boulonnés peuvent provoquer des concentrations de contrainte locale. Les phénomènes mis en jeu sont extrêmement complexes et difficile à modéliser. La pré-charge des pièces d'assemblage souvent mal maîtrisée, le contact frottant essentiel à la tenue mécanique ainsi que de phénomènes locaux rendent compliqué tout dimensionnement précis. Ces pièces très critiques sont donc surdimensionnées pour parer tout imprévu. De façon générale, tout surdimensionnement se paie par une masse globale donc une consommation de carburant plus élevée. L'impact écologique est inévitablement dégradé.

La simulation numérique, bien que devant être soumise à une analyse critique à la fois sur la pertinence des modèles proposés que sur les résultats obtenus, permettrait d'affiner les dimensionnements des pièces et ainsi gagner en masse totale. Qui plus est lors des phases de certifications des appareils des essais de fatigue qui coûtent à la fois du temps et de l'argent sont indispensables. Un recours à une simulation numérique fiable les limiterait et diminuerait alors le temps de conception global d'un nouvel appareil. La simulation numérique de structure mécanique complète et complexe se trouve confrontée à de nombreux problèmes. Premièrement vouloir traiter une structure à une échelle globale en prenant en compte certains détails structuraux implique de gérer un modèle

de très grande taille. En effet comme il est possible de l'observer sur la Figure 1 représentant une partie de fuselage, de nombreux détails comme les hublots, les issues ou encore les longerons et cadres du squelette du fuselage viennent enrichir le modèle et augmenter considérablement la taille du problème une fois discrétisé. Par exemple à la Figure 2 est représenté le modèle discrétisé d'un seul assemblage boulonné. Il devient évident que la taille des problèmes augmente très rapidement à chaque détail pris en compte. De plus les non-linéarités apportent elles aussi leur lot de difficultés au niveau numérique. Des lois non différentiables, des cycles d'hystérésis ou encore des phénomènes très locaux peuvent rendre les méthodes numériques très instables sans garantie d'aboutir à une solution mécaniquement pertinente.

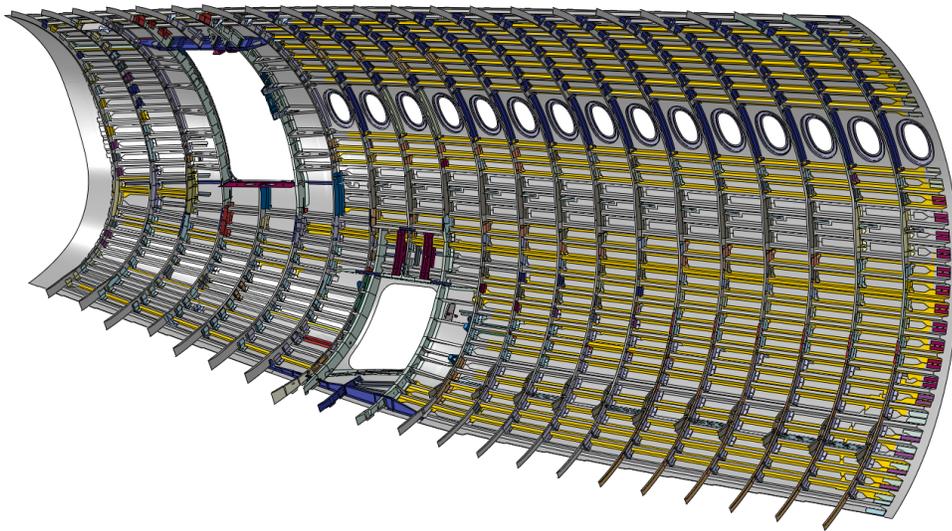


FIGURE 1 – Modèle CAO d'une partie de fuselage - (source AGI)

En plus du temps de résolution qui augmente suivant la taille des modèles, ce sont surtout des limitations en terme de mémoire qui apparaissent et bloquent le plus souvent les résolutions numériques. Même si des bibliothèques (par exemple MUMPS [Amestoy et al., 2001]) permettent de profiter du parallélisme qu'offrent certaines méthodes pour inverser un problème global il est nécessaire de faire appel à d'autres techniques pour dépasser cette limite. En particulier les méthodes de décomposition de domaine permettent de diviser le problème global en sous-problèmes plus petits réduisant ainsi le besoin en mémoire. Nous prenons l'exemple du fuselage d'un A380 sous-structuré (Figure 3) ici en 50 sous-domaines pour répartir la charge en mémoire de chaque sous-structure sur les ordinateurs différents.

Ces méthodes ont été proposées originellement par Schwarz en 1870 [Schwarz, 1870]. Avec l'apparition de l'outil informatique dans la seconde moitié du XX^{ème} siècle, l'intérêt des méthodes de décomposition de domaine s'est fait croissant. Grâce aux nombreuses avancées des deux dernières décennies, les algorithmes deviennent de plus en plus robustes et les problèmes traités de plus en plus variés. Cependant ces études restent souvent dans un cadre linéaire et de nombreux

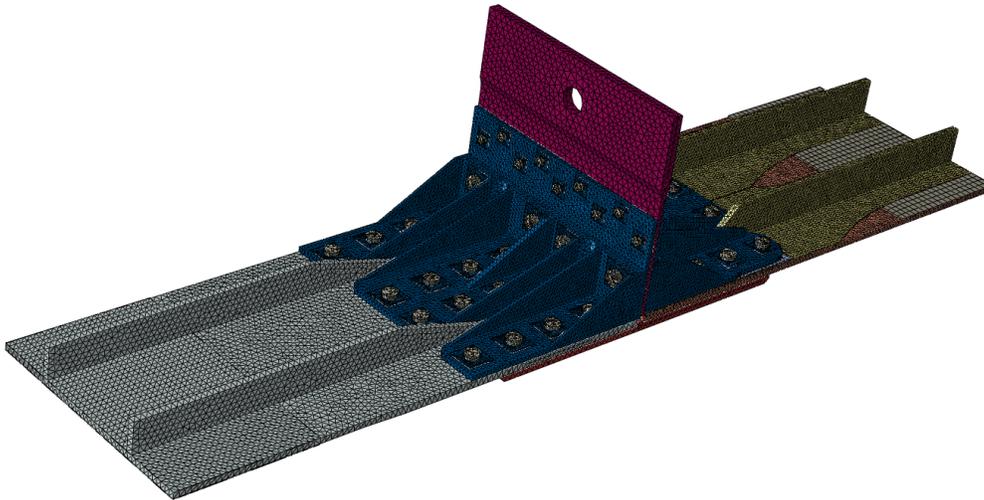


FIGURE 2 – Modèle éléments finis d'un assemblage - (source AGI)

travaux restent encore à mener dans le cadre de la décomposition de domaine non-linéaire. Les non-linéarités, qu'elles soient de type matériaux, géométriques ou encore de contact, sont gérées par des méthodes itératives particulières qui peuvent perturber fortement la convergence des méthodes de décomposition de domaine.

Les méthodes de décomposition de domaine, qu'elles soient non-linéaires ou même linéaires restent dans le giron du monde scientifique académique. Le transfert vers le milieu industriel demeure le maillon faible pour en faire profiter l'ensemble de la communauté scientifique. En plus de la robustesse que doivent avoir ces méthodes pour être applicables à des structures quelconques et des modèles mécaniques variés (élasticité, plasticité, 2D, 3D, coques...) leur caractère extrêmement intrusif ralentit fortement leur développement au sein des codes industriels. Des visions non-intrusives permettent alors de faciliter ce transfert. Il s'agit de proposer des implémentations qui ne nécessitent pas le développement interne de nouvelles fonctionnalités à l'intérieur même des codes industriels. De nombreuses possibilités existent déjà. Par exemple pour Abaqus© ou code_aster [EDF R&D,] une interface python permet de piloter aisément les fonctions internes. L'implémentation non-intrusive se situerait donc au niveau de cette sur-couche, autorisant même un couplage simple avec d'autres outils de calcul scientifique potentiellement plus à même de gérer des non-linéarités particulières.

L'objectif de ces travaux de thèse est de proposer une formulation non-intrusive d'une méthode de décomposition de domaine mixte adaptée au calcul d'assemblage. Cela consiste à adapter la méthode Latin avec décomposition de domaine [Ladevèze, 1999] pour l'implémenter dans code_aster. Afin de présenter ces travaux le manuscrit se sous-structure en cinq chapitres :

- Le [premier chapitre](#) présente des éléments bibliographique ayant trait à la décomposition de domaine et la gestion du contact au sein de ces méthodes parallèles. Une première partie

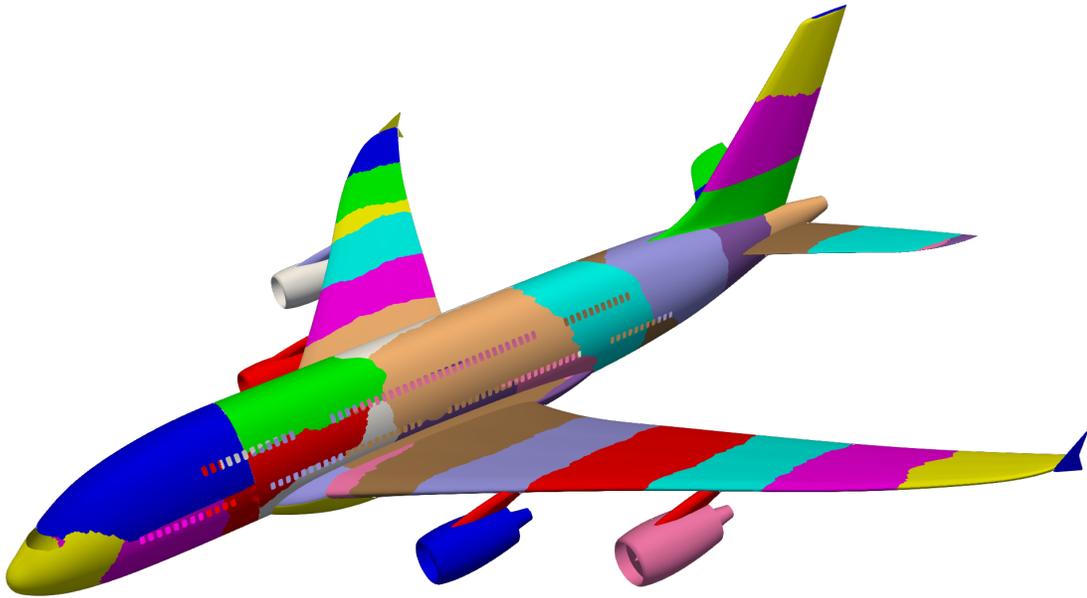


FIGURE 3 – Exemple de sous-structuration sur un fuselage d’A380 - géométrie obtenue sur [GrabCAD,]

aborde les méthodes de décomposition de domaine telles que Schwarz [Gander, 2006], BDD [Mandel, 1993] ou encore FETI [Farhat and Roux, 1991] dans un cadre simple d’élasticité linéaire. Une deuxième partie détaille les techniques les plus employées pour gérer un problème de contact. Les techniques de pénalisation ou lagrangien couplées à des méthodes de type statuts, Uzawa ou Newton [Yastrebov, 2011] permettent de prendre en compte des lois de contact frottant. Bien que développées depuis de nombreuses années, les méthodes de décomposition de domaine n’ont inclus que très récemment la gestion du contact [Dureisseix and Farhat, 2001, Dostál and Horák, 2004]. Ces aspects sont présentés au sein d’une troisième partie. Plus particulièrement sont détaillés les travaux de Dostal [Dostál et al., 2005] sur la méthode FETI.

- Le deuxième chapitre détaille la méthode Latin [Ladevèze, 1999] qui est un solveur adapté aux problèmes non-linéaires. Le premier principe de séparer les équations en deux groupes est utilisé pour résoudre un problème de décomposition de domaine. Les sous-domaines sont considérés de manière distincte des interfaces. Le deuxième principe fait intervenir deux directions de recherche afin de mettre en place un algorithme itératif. On résout successivement un problème sur les interfaces puis un problème sur les sous-domaines. Les directions de recherche introduisent des conditions mixtes d’interface et rendent alors l’implémentation de l’algorithme très délicate au sein de codes industriels. Nous proposons donc d’introduire une nouvelle direction de recherche non-locale s’appuyant sur l’ajout d’une couche d’élé-

ment au niveau de l'interface. Cela permet de transformer des manipulations très intrusives en opérations élémentaires implémentées dans l'ensemble des codes industriels. Cette nouvelle direction de recherche permet de coupler les noeuds d'interface entre eux et assure une définition correcte des efforts d'interface. Au sein de ce chapitre sont considérées seulement des interfaces parfaites. Une validation de l'implémentation non-intrusive par rapport à une implémentation classique est présentée sur un premier cas simple 1D. Une structure 3D est ensuite étudiée et nous proposons une comparaison à un calcul de référence effectué directement dans *code_aster*. Pour ce cas ci, nous considérons une sous-structuration automatique de la structure.

- Le [troisième chapitre](#) vise à présenter plus précisément la gestion de l'étape linéaire. Après une introduction qui illustre les limites d'une approche mono-échelle, l'extensibilité de la méthode fait l'objet de la deuxième section. Il s'avère que l'aspect non-intrusif ne modifie en rien la formulation multi-échelle classique de la méthode Latin [[Ladevèze et al., 2001](#)]. Le déséquilibre des champs d'effort à l'étape linéaire est compensé par l'ajout d'un problème macroscopique. Une vérification globale de l'équilibre des efforts d'interface est alors assurée. Une troisième section se penche sur la discontinuité des déplacements d'interface. Avec la même idée que pour les efforts, une vérification globale de la continuité des déplacements est introduite. Une quatrième section présente les comparaisons des deux approches multi-échelle. Nous montrons en particulier que cette approche multi-échelle en continuité ne permet pas de garantir l'extensibilité. La cinquième section du chapitre s'intéresse à une accélération de type Krylov pour remplacer l'actuelle relaxation à la fin de l'étape linéaire. L'idée est d'inclure les itérations de la méthode Latin au sein d'un algorithme de Krylov permettant d'optimiser les solutions itératives par rapport à celles déterminées précédemment. Cela accélère en particulier la convergence de la méthode.
- Le [quatrième chapitre](#) a pour objectif de détailler le traitement de l'étape locale. Les lois de contact par nature extrêmement locales nous conduisent, par soucis d'implémentation, à simplifier l'opérateur non-intrusif de direction de recherche afin de découpler les comportements de chaque noeud de l'interface. Une proposition de découplage de la direction de recherche à partir de l'opérateur non-local est présentée. Nous détaillons ensuite la gestion du contact frottant. Les directions de recherche étant liées aux sous-domaines, il convient de proposer une formulation qui prenne en compte ces directions de recherche différentes de chaque côté des interfaces. La pré-charge des éléments d'assemblage étant gérée par une interface de type contact, il est aussi détaillé l'implémentation de ce type de comportement. Des validations par rapport à des solutions de références et des travaux antérieurs sont présentées.
- Le [cinquième chapitre](#) aborde des aspects implémentations et performances de la méthode. Une première section s'attache à présenter les principes et la structure de *code_aster*. Nous expliquons dans une deuxième section l'implémentation de l'outil que nous avons développé au cours de ces travaux de thèse. Des détails sur la phase de mise en donnée et de préparation, ainsi que sur le code Latin non-intrusif sont exposés. Une troisième section expose des cas académiques "extrêmes" : étude du *speed-up*, grand nombre d'interface de contact. Dans une quatrième section des structures industrielles sont simulées : une première étude sur le comportement "cyclique" d'une bride assemblée aéronautique, une seconde structure fournie par

Airbus Group impliquant de nombreuses pré-charges est re-décomposée automatiquement pour diminuer le temps de calcul.

État de l'art du traitement du contact en décomposition de domaine

Ce chapitre a pour objectif de présenter un état de l'art du traitement du contact au sein des méthodes de décomposition de domaine. Ainsi sont présentées successivement les différentes approches de décomposition de domaine dans le cadre de l'élasticité linéaire sous l'hypothèse des petites déformations, le traitement général d'un problème de contact, et enfin la gestion du contact au sein des méthodes de décomposition de domaine.

Sommaire

1	Méthodes de Schwarz continue	9
1.1	Méthode de Schwarz alternée	10
1.2	Méthode de Schwarz parallèle	11
1.3	Méthode de Schwarz sans recouvrement	12
2	Méthodes de Schwarz discrètes	13
2.1	La Multiplicative Schwarz method	14
2.2	La Additive de Schwarz Method	15
2.3	La restricted additive Schwarz method	16
3	Approches primales et duales sans recouvrement	17

3.1	Généralités pour les problèmes sans recouvrements - notations	17
3.1.1	Sous-structuration sans recouvrement	17
3.1.2	Généralisation à plusieurs sous-domaines - notations bloc	18
3.1.3	Opérateurs de somme et différence d'interface	19
3.1.4	Problème sous-structuré	20
3.1.5	Opérateurs et problèmes de Schur	20
3.2	Approches primales	22
3.2.1	Problème d'interface	22
3.2.2	Choix du préconditionneur	22
3.2.3	Problème grossier	23
3.2.4	Illustration de l'approche primale	24
3.3	Approches duales	24
3.3.1	Problème d'interface	24
3.3.2	Choix du préconditionneur	25
3.3.3	Problème grossier	26
3.3.4	Autres versions de la méthode FETI	26
4	Généralités sur le traitement du contact frottant	28
4.1	Formulation du problème de contact frottant	28
4.2	Boucle interne : prise en compte des conditions limites de contact	30
4.2.1	Pénalisation	30
4.2.2	Lagrangien augmenté	30
4.2.3	Méthode de Nitsche	31
4.3	Boucle externe : algorithme d'optimisation	33
4.3.1	L'algorithme Partial Dirichlet-Neumann ou méthode des statuts	33
4.3.2	Algorithme de Uzawa	35
4.3.3	Méthodes de Newton	36
5	Prise en compte du contact en décomposition de domaine	37
5.1	FETI pour le contact	37
5.1.1	Définition du problème et notations	38
5.1.2	SMALBE : Semi Monotonic Augmented Lagrangian for Bound and Equality constrained QP problems	39
5.1.3	Minimisation sous contrainte	40
5.2	Méthodes de Schwarz pour le contact	43
6	Bilan	43

Problème mécanique et notations

Avant de présenter différentes méthodes de décomposition de domaine il convient d'introduire le problème mécanique étudié par la suite ainsi que quelques notations. Nous considérons un problème mécanique d'élasticité linéaire isotrope caractérisé par le tenseur de Hooke associé et noté \mathbb{K} .

La structure étudiée occupe un domaine Ω et elle est soumise à des conditions limites de Dirichlet u_d sur le bord $\partial_u\Omega$ ainsi que des conditions limites de Neumann F_d sur le bord complémentaire $\partial_F\Omega$. Un effort volumique f_d défini sur Ω est aussi pris en compte. La formulation forte du problème est :

Trouver (u, σ) tel que :

$$\begin{cases} \operatorname{div}\sigma + f_d = 0 & \text{sur } \Omega & \text{équilibre} \\ \sigma = \mathbb{K} : \epsilon & \text{sur } \Omega & \text{loi de comportement} \\ \epsilon = \frac{1}{2}(\operatorname{grad}u + \operatorname{grad}^T u) & \text{sur } \Omega & \text{HPP} \\ u = u_d & \text{sur } \partial_u\Omega & \text{CL de Dirichlet} \\ \sigma \cdot n = F_d & \text{sur } \partial_F\Omega & \text{CL de Neumann} \end{cases} \quad (1.1)$$

Sous forme faible le problème consiste à trouver $u \in \mathcal{U}$ tel que $\forall v \in \mathcal{U}_0$:

$$\begin{cases} \int_{\Omega} \sigma : \epsilon(v) d\Omega = \int_{\Omega} f_d \cdot v d\Omega + \int_{\partial_F\Omega} F_d \cdot v dS \\ \sigma = \mathbb{K}\epsilon(u) \end{cases} \quad (1.2)$$

avec $\mathcal{U} = \{u \in H^1(\Omega), u = u_d \text{ sur } \partial_u\Omega\}$ l'espace des déplacements cinématiquement admissibles et $\mathcal{U}_0 = \{u \in H^1(\Omega), u = 0 \text{ sur } \partial_u\Omega\}$ l'espace des déplacements cinématiquement admissibles à 0. Cette formulation faible du problème nous permet d'introduire la version discrétisée utilisant une méthode éléments finis classique. En considérant ϕ la matrice des fonctions de forme éléments finis et \mathbf{u} le champ de déplacement discrétisé :

$$u = \phi \mathbf{u} \quad (1.3)$$

Dans ce cas le problème discrétisé s'écrit classiquement :

$$\mathbf{K} \mathbf{u} = \mathbf{f}_d \quad (1.4)$$

avec \mathbf{K} l'opérateur de rigidité discrétisé prenant en compte les conditions limites en déplacement.

1 Méthodes de Schwarz continue

Nous introduisons ici le principe des méthodes de décomposition de domaine à travers la formulation continue des méthodes de Schwarz. Lorsque Schwarz en 1870 [Schwarz, 1870] a introduit la notion de décomposition de domaine, il était motivé par le fait de déterminer la solution d'une équation aux dérivées partielles sur un domaine de forme non triviale. Ne pouvant exhiber de solutions analytiques que pour des domaines de forme particulières : disque et rectangle, il eut l'idée de décomposer le domaine non trivial en domaines élémentaires pour résoudre l'équation de façon séparée (Figure 1.1). Bien entendu tout l'enjeu est de faire le lien entre les solutions calculées sur les sous-domaines afin de reconstruire la solution sur l'ensemble du domaine.

Dans cette première section le problème que nous chercherons à résoudre est : connaissant g défini sur $\partial\Omega$ trouver u sur $\Omega = \Omega_1 \cup \Omega_2$ tel que :

$$\begin{cases} \Delta u = 0 & \text{sur } \Omega \\ u = g & \text{sur } \partial\Omega \end{cases} \quad (1.5)$$

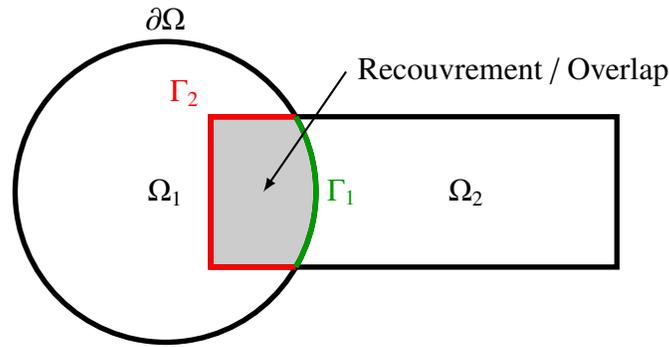


FIGURE 1.1 – Décomposition de Schwarz

1.1 Méthode de Schwarz alternée

Suivant la Figure 1.1, le domaine Ω est décomposé en deux sous-structures Ω_1 circulaire et Ω_2 rectangulaire. On notera : Γ_1 la partie du bord de Ω_1 incluse dans Ω_2 , Γ_2 la partie du bord de Ω_2 incluse dans Ω_1 . Ainsi :

$$\begin{aligned}\Gamma_1 &= \partial\Omega_1 \cap \Omega_2 \\ \Gamma_2 &= \partial\Omega_2 \cap \Omega_1\end{aligned}\quad (1.6)$$

$\Omega_1 \cap \Omega_2$ est appelé recouvrement ou *overlap*. L'idée de la méthode de Schwarz alternée est de chercher successivement une solution dans Ω_1 puis Ω_2 comme décrit à l'algorithme 1.1 et illustré sur la Figure 1.2.

Algorithme 1.1 : Méthode de Schwarz alternée

Input : Initialisation : $u_2^0 = 0$

for $n = 0, 1 \dots$ **do**

 Trouver u_1^{n+1} tel que :

$$\begin{aligned}\Delta u_1^{n+1} &= 0 && \text{sur } \Omega_1 \\ u_1^{n+1} &= g && \text{sur } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} &= u_2^n && \text{sur } \Gamma_1\end{aligned}\quad (1.7)$$

 Trouver u_2^{n+1} tel que :

$$\begin{aligned}\Delta u_2^{n+1} &= 0 && \text{sur } \Omega_2 \\ u_2^{n+1} &= g && \text{sur } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} &= u_1^{n+1} && \text{sur } \Gamma_2\end{aligned}\quad (1.8)$$

Après avoir initialisé u_2^0 à 0, on effectue de manière itérative un premier problème sur Ω_1 pour déterminer u_1^{n+1} avec en conditions limites : g sur le bord commun de $\partial\Omega_1$ et $\partial\Omega$, et la solution calculée précédemment u_2^n sur Γ_1 . Le deuxième problème à résoudre consiste à déterminer u_2^{n+1} sur Ω_2 avec pour conditions limites : g sur $\partial\Omega_2 \cap \partial\Omega$ et u_1^{n+1} sur Γ_2 .

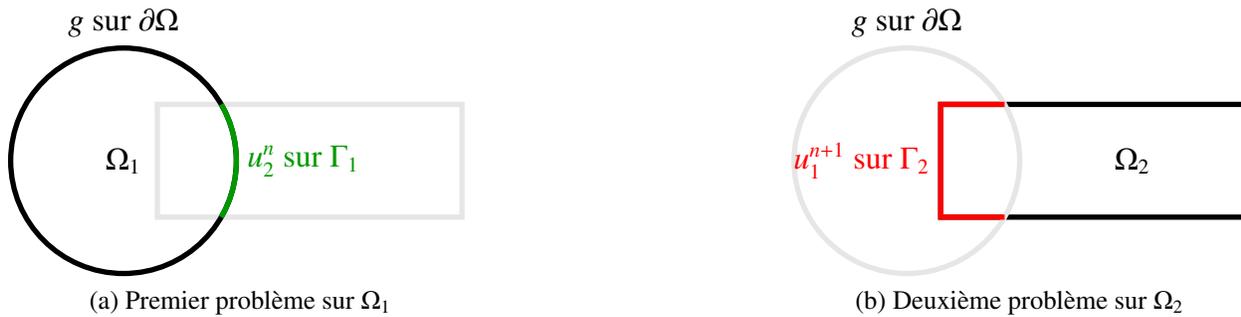


FIGURE 1.2 – Problèmes alternés de Schwarz

⚠ Remarque 1.0

La convergence est pilotée par le choix du recouvrement. Plus ce recouvrement est important plus la convergence de l'algorithme sera rapide. La méthode de Schwarz alternée est généralisable à plusieurs sous-domaines. Cependant il est possible que certains sous-domaines soient reliés à plusieurs autres et il convient de faire attention à l'ordre dans lequel résoudre les problèmes sur les sous-domaines.

L'inconvénient de la méthode de Schwarz alternée réside dans le fait qu'il est nécessaire de résoudre le problème sur un sous-domaine avant de pouvoir résoudre le problème sur l'autre. Elle ne peut donc pas être parallélisée. Cependant dans le cas de multiples sous-domaines, des techniques basées sur du coloriage [Saad and Sosonkina, 1999] permettent de paralléliser les sous-domaines qui n'ont pas de recouvrement en commun.

1.2 Méthode de Schwarz parallèle

Nous avons rappelé précédemment la méthode de Schwarz alternée. P.-L. Lions a proposé une version totalement parallèle [Lions, 1988] qui permet de profiter de la puissance de l'outil informatique pour gagner en rapidité de calcul. Ainsi au lieu d'effectuer des calculs successifs à chaque itération, il est effectué des calculs en parallèle sur chaque sous-domaine. Cela se traduit simplement par un glissement d'indice dans l'algorithme 1.2. Les solutions u_1^n et u_2^n connues à l'itération n sont utilisées en parallèle pour déterminer u_1^{n+1} et u_2^{n+1} .

⚠ Remarque 1.1

La version parallèle peut elle aussi être généralisée à plus de deux sous-domaines. Cependant le cas de recouvrements multiples peut poser des problèmes de convergence. En effet contrairement à la version alternée, il peut être beaucoup plus compliqué de trouver la bonne séquence pour appliquer les conditions limites dans les recouvrements multiples. Pour cela Lions [Lions, 1988]

propose tout simplement d'éviter les recouvrements multiples (Figure 1.3). C'est-à-dire que :

$$\text{Si } \Omega_i \cap \Omega_j \neq \emptyset \text{ et } \Omega_i \cap \Omega_k \neq \emptyset \text{ alors } \Omega_j \cap \Omega_k = \emptyset \quad (1.9)$$

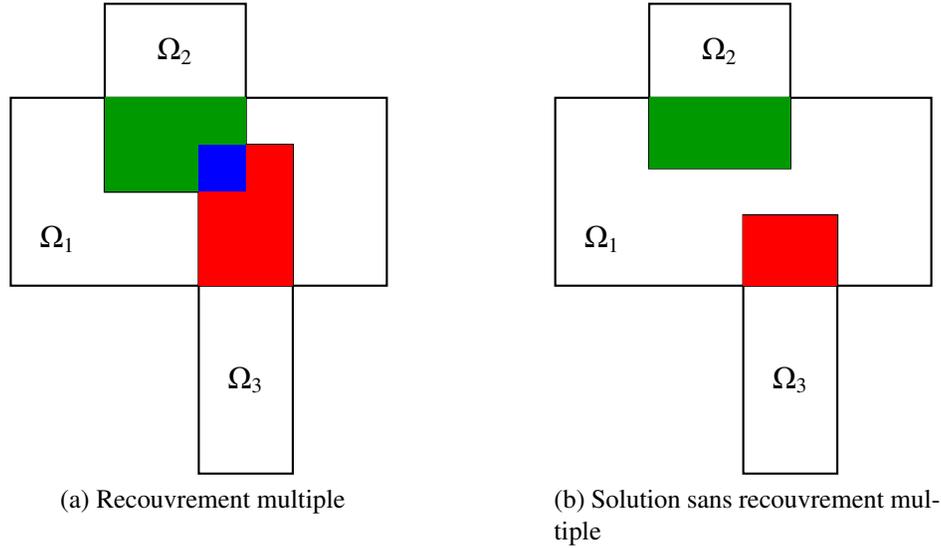


FIGURE 1.3 – Problème de recouvrements multiples

Algorithme 1.2 : Méthode de Schwarz parallèle

Input : Initialisation : $u_2^0 = 0$

for $n = 0, 1 \dots$ **do**

Trouver u_1^{n+1} et u_2^{n+1} tels que :

$$\begin{array}{ll}
 \Delta u_1^{n+1} = 0 & \text{sur } \Omega_1 \\
 u_1^{n+1} = g & \text{sur } \partial\Omega_1 \cap \partial\Omega \\
 u_1^{n+1} = u_2^n & \text{sur } \Gamma_1 \\
 \Delta u_2^{n+1} = 0 & \text{sur } \Omega_2 \\
 u_2^{n+1} = g & \text{sur } \partial\Omega_2 \cap \partial\Omega \\
 u_2^{n+1} = u_1^n & \text{sur } \Gamma_2
 \end{array} \quad (1.10)$$

1.3 Méthode de Schwarz sans recouvrement

Dans le contexte des assemblages, le recouvrement des méthodes de Schwarz s'oppose à l'idée de distinguer des sous-domaines distincts par pièces et indépendants des autres sous-domaines. Ce recouvrement absolument nécessaire pour converger limite alors les cas d'étude. Pour cela Lions a proposé une version des méthodes de Schwarz sans recouvrement [Lions, 1990] (Figure 1.4). Dans le cas du recouvrement il suffit d'utiliser la variable primale de déplacement u . En effet l'égalité du déplacement entre deux sous-domaines dans l'*overlap* implique l'égalité du gradient de déplacement ∇u . Les grandeurs duales liées au gradient de u convergent donc implicitement. Sans

recouvrement cette propriété disparaît et de nouvelles conditions d'interface doivent être définies. Pour cela des conditions de Robin sont introduites sur l'interface entre deux sous-domaines et permettent de transmettre les informations duales. Ces conditions de Robin sont caractérisées par les paramètres p_1 et p_2 . L'algorithme est décrit à l'algorithme 1.3.

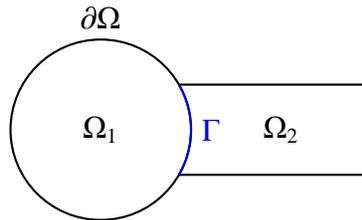


FIGURE 1.4 – Exemple d'une décomposition sans recouvrement

⚠ Remarque 1.2

Les méthodes de Schwarz dites optimisées sont basées sur la même idée d'introduire des conditions de Robin au niveau des interfaces. Les conditions de transmission généralisées dépendent du type d'équations aux dérivées partielles qui est traité. De nombreuses références et propositions d'opérateurs optimaux peuvent être trouvées dans [Gander, 2006, Gander et al., 2008]. Une application à des milieux hétérogènes est proposée par [Maday and Magoulès, 2007]. Une adaptation pour des problèmes de Helmholtz est présentée par [Magoulès et al., 2004a, Gander et al., 2007].

Bien entendu cette version sans recouvrement est généralisable à plus de deux sous-domaines.

2 Méthodes de Schwarz discrètes

Après avoir présenté les formulations continues des méthodes de Schwarz, nous présentons dans cette section les formulations discrètes adaptées aux méthodes éléments finis. En particulier nous abordons la *méthode de Schwarz multiplicative*, la *méthode de Schwarz additive* ou encore la *méthode Restricted Additive Schwarz*. Pour ce faire nous nous focalisons sur le système 1.4 :

$$\mathbf{K}\mathbf{u} = \mathbf{f}_d \quad (1.13)$$

avec \mathbf{K} l'opérateur discrétisé caractérisant l'équation du problème, \mathbf{u} le champ de déplacement discrétisé et \mathbf{f}_d le vecteur de chargement. Afin de présenter les méthodes de Schwarz discrétisées nous introduisons les opérateurs de restrictions $(\mathbf{R}_i)_{i=1..N}$ relatifs à chaque sous-domaine (Figure 1.5). Avec ces opérateurs nous extrayons du champ éléments finis global \mathbf{u} les champs de déplacement éléments finis sur chaque sous-domaine : $(\mathbf{u}_i = \mathbf{R}_i\mathbf{u})_{i=1..N}$.

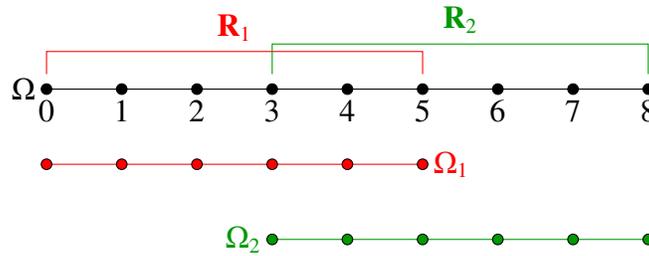
Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages

Algorithme 1.3 : Méthode de Schwarz alternée sans recouvrement**Input** : Initialisation : $u_2^0 = 0$ **for** $n = 0, 1 \dots$ **do**Trouver u_1^{n+1} tel que :

$$\begin{aligned} \Delta u_1^{n+1} &= 0 && \text{sur } \Omega_1 \\ u_1^{n+1} &= g && \text{sur } \partial\Omega_1 \cap \partial\Omega \\ (\partial_{n_1} + p_1) u_1^{n+1} &= (\partial_{n_1} + p_1) u_2^n && \text{sur } \Gamma_1 \end{aligned} \quad (1.11)$$

Trouver u_2^{n+1} tel que :

$$\begin{aligned} \Delta u_2^{n+1} &= 0 && \text{sur } \Omega_2 \\ u_2^{n+1} &= g && \text{sur } \partial\Omega_2 \cap \partial\Omega \\ (\partial_{n_2} + p_2) u_2^{n+1} &= (\partial_{n_2} + p_2) u_1^{n+1} && \text{sur } \Gamma_2 \end{aligned} \quad (1.12)$$



$$\mathbf{R}_1 = \begin{pmatrix} 1 & & & & 0 & 0 & 0 \\ & 1 & & & 0 & 0 & 0 \\ & & 1 & & 0 & 0 & 0 \\ & & & 1 & 0 & 0 & 0 \\ & & & & 1 & 0 & 0 \\ & & & & & 1 & 0 & 0 \\ & & & & & & 1 & 0 & 0 \end{pmatrix} \quad \mathbf{R}_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & & & & & \\ 0 & 0 & 0 & & 1 & & & & \\ 0 & 0 & 0 & & & 1 & & & \\ 0 & 0 & 0 & & & & 1 & & \\ 0 & 0 & 0 & & & & & 1 & \\ 0 & 0 & 0 & & & & & & 1 \\ 0 & 0 & 0 & & & & & & & 1 \end{pmatrix} \quad (1.14)$$

FIGURE 1.5 – Exemple de sous-structuration et opérateurs de restriction

Cela permet de définir par la même occasion les restrictions de l'opérateur de rigidité \mathbf{K} aux sous-domaines :

$$\begin{aligned} \mathbf{K}_1 &= \mathbf{R}_1 \mathbf{K} \mathbf{R}_1^T \\ \mathbf{K}_2 &= \mathbf{R}_2 \mathbf{K} \mathbf{R}_2^T \end{aligned} \quad (1.15)$$

2.1 La Multiplicative Schwarz method

La *multiplicative Schwarz method* [Chan and Mathew, 1994, Smith et al., 2004] pourrait être qualifiée de version discrétisée de la méthode de Schwarz alternée continue. En effet on résout en

deux étapes un premier problème sur le sous-domaine 1 et un second problème sur le sous-domaine 2 prenant en compte les résultats du calcul précédent. Cela se traduit par l'algorithme 2.1.

Algorithme 2.1 : Méthode de Schwarz multiplicative

Input : Initialisation : $\mathbf{u}^0 = 0$

for $n = 0, 1 \dots$ **do**

 Trouver $\mathbf{u}^{n+1/2}$ tel que :

$$\mathbf{u}^{n+1/2} = \mathbf{u}^n + \mathbf{R}_1^T \mathbf{K}_1^{-1} \mathbf{R}_1 (\mathbf{f}_d - \mathbf{K} \mathbf{u}^n) \quad (1.16)$$

 Trouver \mathbf{u}^{n+1} tel que :

$$\mathbf{u}^{n+1} = \mathbf{u}^{n+1/2} + \mathbf{R}_2^T \mathbf{K}_2^{-1} \mathbf{R}_2 (\mathbf{f}_d - \mathbf{K} \mathbf{u}^{n+1/2}) \quad (1.17)$$

 **Remarque 1.3**

Dans le cas de non recouvrement, la *multiplicative Schwarz method* est équivalente à une méthode bloc de Gauss-Seidel.

Une extension à des problèmes non symétriques peut être trouvée dans [Cai and Widlund, 1993].

2.2 La Additive de Schwarz Method

La méthode de Schwarz multiplicative a le même inconvénient que son pendant continu. Cette méthode séquentielle n'est pas particulièrement adaptée à la décomposition de domaine massivement parallèle. La méthode de Schwarz additive [Dryja and Widlund, 1989, Dryja and Widlund, 1990] qui peut être considérée comme une version discrète d'une version parallèle continue particulière proposée dans [Matsokin A. M., 1985] permet une parallélisation des résolutions sur les sous-domaines. Pour cela Dryja et Widlung [Dryja and Widlund, 1994] ont proposé de préconditionner le problème global par $P_{AS}^{-1} = \mathbf{R}_1^T \mathbf{K}_1^{-1} \mathbf{R}_1 + \mathbf{R}_2^T \mathbf{K}_2^{-1} \mathbf{R}_2$:

$$\left(\mathbf{R}_1^T \mathbf{K}_1^{-1} \mathbf{R}_1 + \mathbf{R}_2^T \mathbf{K}_2^{-1} \mathbf{R}_2 \right) \mathbf{K} \mathbf{u} = \left(\mathbf{R}_1^T \mathbf{K}_1^{-1} \mathbf{R}_1 + \mathbf{R}_2^T \mathbf{K}_2^{-1} \mathbf{R}_2 \right) \mathbf{f}_d \quad (1.18)$$

Généralisé à des multiples sous-domaines :

$$P_{AS}^{-1} = \sum_{i=1}^N \mathbf{R}_i^T \mathbf{K}_i^{-1} \mathbf{R}_i \quad (1.19)$$

Ce préconditionnement conduit en particulier à l'algorithme itératif 2.2.

Algorithme 2.2 : Méthode de Schwarz additive**Input** : Initialisation : $\mathbf{u}^0 = 0$ **for** $n = 0, 1 \dots$ **do**Trouver \mathbf{u}^{n+1} tel que :

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \left(\underbrace{\mathbf{R}_1^T \mathbf{K}_1^{-1} \mathbf{R}_1}_{\text{Résolution sur } \Omega_1} + \underbrace{\mathbf{R}_2^T \mathbf{K}_2^{-1} \mathbf{R}_2}_{\text{Résolution sur } \Omega_2} \right) (\mathbf{f}_d - \mathbf{K} \mathbf{u}^n) \quad (1.20)$$

! Remarque 1.4

- Avec recouvrement la méthode de Schwarz additive est équivalente à la version discrète de la méthode parallèle en continue, sans recouvrement elle correspond à une méthode bloc de Jacobi.
- Les formulations présentées jusqu'ici restent des algorithmes de points fixes dont la convergence est lente. Il est possible de les inclure dans un algorithme de Krylov et ainsi en accélérer la convergence. En particulier, puisque le préconditionnement proposé dans le cas de la méthode additive est symétrique, une méthode de gradient conjugué peut être utilisée.

La convergence est fortement liée au nombre de sous-domaines. Puisque les sous-domaines ne communiquent qu'avec leurs voisins, le nombre total de communications entre sous-domaines (et donc d'itérations) augmente quand le nombre de sous-domaines augmente lui aussi. Cette formulation n'est pas extensible. L'extensibilité est alors obtenue via l'introduction d'un problème grossier dans le préconditionneur qui devient :

$$P_{AS_c}^{-1} = \sum_{i=0}^N \mathbf{R}_i^T \mathbf{K}_i^{-1} \mathbf{R}_i + \underbrace{\mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{R}_0}_{\text{correction grossière}} \quad (1.21)$$

Des comparaisons du taux de convergence entre la méthode additive et multiplicative peuvent être trouvées dans [Griebel and Oswald, 1995, Nabben, 2003].

2.3 La restricted additive Schwarz method

La *restricted additive Schwarz method* est une adaptation proposée par [Cai and Sarkis, 1999] de la version additive. Le nouveau préconditionneur permet de réduire le nombre d'échanges entre les sous-domaines. Il a même été noté par Cai et Sarkis que le nombre d'itérations à convergence se trouvait réduit par rapport à une version additive. Le préconditionneur de la Restricted Additive Schwarz method est :

$$P_{RAS}^{-1} = \sum_{i=0}^N \tilde{\mathbf{R}}_i^T \mathbf{K}_i^{-1} \mathbf{R}_i \quad (1.22)$$

Avec $\tilde{\mathbf{R}}_i$ un nouvel opérateur de restriction qui se référerait à des sous-domaines $\tilde{\Omega}_i$ sans recouvrement.

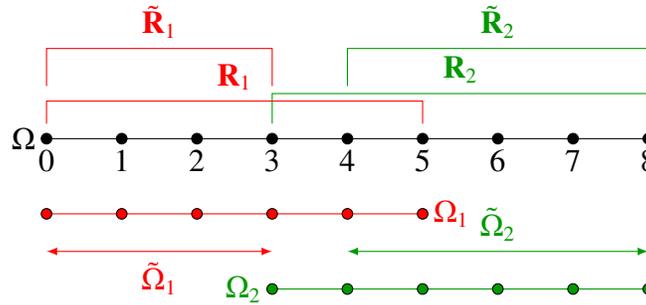


FIGURE 1.6 – Exemple d’opérateur de restriction $\tilde{\mathbf{R}}$

Algorithme 2.3 : Restricted Additive Schwarz method

Input : Initialisation : $\mathbf{u}^0 = 0$

for $n = 0, 1 \dots$ **do**

 Trouver \mathbf{u}^{n+1} tel que :

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{i=0}^N \tilde{\mathbf{R}}_i^T \mathbf{K}_i^{-1} \mathbf{R}_i (\mathbf{f}_d - \mathbf{K}\mathbf{u}^n) \quad (1.23)$$

3 Approches primales et duales sans recouvrement

3.1 Généralités pour les problèmes sans recouvrements - notations

Comme indiqué lors de la présentation des méthodes de Schwarz, il est important de pouvoir gérer des méthodes de décomposition de domaine sans recouvrement. Cela permet de manipuler plus aisément des problèmes hétérogènes. De plus dans l’optique de traiter des assemblages avec des interactions particulières entre pièces, il peut être judicieux de chercher à distinguer les différentes pièces par des sous-domaines distincts. C’est pourquoi nous présentons désormais quelques techniques de décomposition de domaine sans recouvrement. Contrairement aux méthodes de Schwarz avec recouvrement pour lesquelles l’overlap permet de converger seulement grâce à des contraintes sur le champ de déplacement, il est ici nécessaire de vérifier la continuité des déplacements ainsi que l’équilibre des efforts aux interfaces.

3.1.1 Sous-structuration sans recouvrement

Un exemple de sous-structuration est illustré à la Figure 1.7. Un domaine Ω est divisé en deux sous-domaines Ω_1 et Ω_2 qui échangent au niveau de l’interface Γ_{12} respectivement des champs

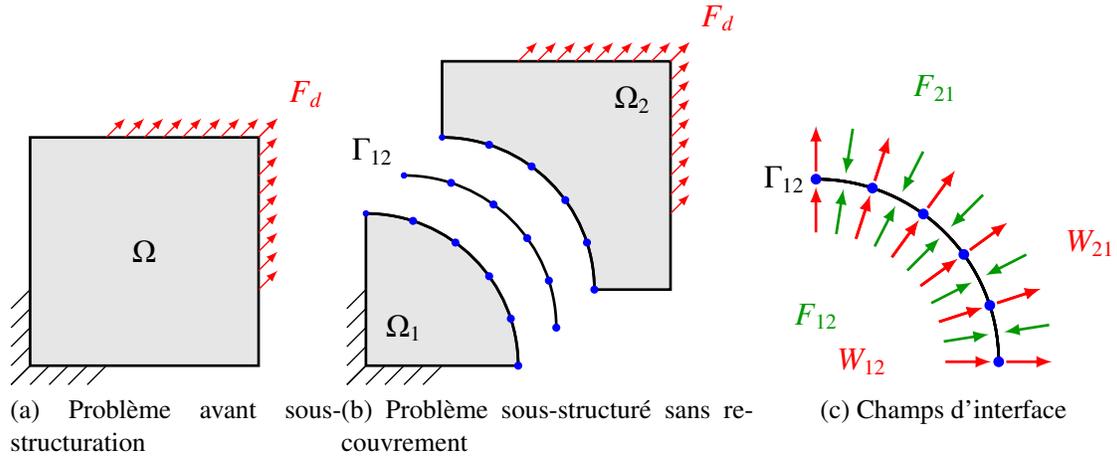


FIGURE 1.7 – Exemple de sous-structuration

d'interface de déplacement et d'effort $(\mathbf{W}_{12}, \mathbf{F}_{12})$ et $(\mathbf{W}_{21}, \mathbf{F}_{21})$. Le champ d'effort \mathbf{F}_{12} (respectivement \mathbf{F}_{21}) est défini comme l'action de Ω_1 sur Ω_2 (respectivement de Ω_2 sur Ω_1). Afin de vérifier les conditions d'interface, les champs de déplacement doivent être continus et les champs d'effort équilibrés. Nous définissons respectivement \mathbf{N}_{12} et \mathbf{N}_{21} les opérateurs de trace des sous-domaines Ω_1 et Ω_2 relativement à l'interface Γ_{12} . Ces opérateurs permettent de transférer un champ défini sur l'ensemble des noeuds d'un sous-domaine à la restriction de ce champ aux noeuds de l'interface considérée. Par exemple :

$$\begin{aligned} \mathbf{W}_{12} &= \mathbf{N}_{12} \mathbf{u}_1 \\ \mathbf{W}_{21} &= \mathbf{N}_{21} \mathbf{u}_2 \end{aligned} \quad (1.24)$$

⚠ Remarque 1.5

L'opérateur \mathbf{N}_{12}^T permet de prolonger par 0 un champ défini sur les noeuds de l'interface Γ_{12} à un champ défini sur l'ensemble du sous-domaine Ω_1

3.1.2 Généralisation à plusieurs sous-domaines - notations bloc

Pour un cas général, nous noterons \mathcal{E} l'ensemble regroupant les différents sous-domaines et \mathcal{G} l'ensemble regroupant les interfaces. Si deux sous-domaines $(\Omega_i, \Omega_j) \in \mathcal{E}$ partagent une interface $\Gamma_{ij} \in \mathcal{G}$, nous noterons respectivement $(\mathbf{W}_{ij}, \mathbf{F}_{ij})$ et $(\mathbf{W}_{ji}, \mathbf{F}_{ji})$ les grandeurs d'interface que s'échangent les sous-domaines Ω_i et Ω_j . Les conditions d'interface imposent alors :

$$\begin{aligned} \mathbf{W}_{ij} - \mathbf{W}_{ji} &= 0 \\ \mathbf{F}_{ij} + \mathbf{F}_{ji} &= 0 \end{aligned} \quad (1.25)$$

Dans le cas de nombreuses interfaces par sous-domaine, nous définissons les grandeurs d'interface concaténées relatives à Ω_i :

$$\mathbf{W}_i = \begin{pmatrix} \vdots \\ \mathbf{W}_{ij} \\ \vdots \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \vdots \\ \mathbf{F}_{ij} \\ \vdots \end{pmatrix}, \quad \mathbf{N}_i = \begin{pmatrix} \ddots & & \\ & \mathbf{N}_{ij} & \\ & & \ddots \end{pmatrix}, \quad j \text{ tel que } \Omega_j \text{ voisin de } \Omega_i \quad (1.26)$$

Avec la même idée, nous utilisons les grandeurs bloc qui permettent de concaténer les notations précédentes relatives aux sous-domaines. Si \mathbf{x}_i représente une grandeur définie au niveau des sous-structures, \mathbf{x} représente la concaténation globale sur l'ensemble des sous-domaines.

$$\mathbf{x} = \begin{pmatrix} \vdots \\ \mathbf{x}_i \\ \vdots \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} \ddots & & 0 \\ & \mathbf{K}_i & \\ 0 & & \ddots \end{pmatrix}, \quad \text{où } i \in \mathcal{E} \quad (1.27)$$

3.1.3 Opérateurs de somme et différence d'interface

Nous introduisons l'opérateur \mathbf{A} de somme des vecteurs d'interface et l'opérateur \mathbf{B} de différence qui sont définis comme suit :

$$\begin{aligned} (\mathbf{A}\mathbf{F})_{\Gamma_{ij}} &= \mathbf{F}_{ij} + \mathbf{F}_{ji} \\ (\mathbf{B}\mathbf{W})_{\Gamma_{ij}} &= \mathbf{W}_{ji} - \mathbf{W}_{ij} \end{aligned} \quad (1.28)$$

Remarque 1.6

\mathbf{A} et \mathbf{B} s'appliquent à des grandeurs définies sur les bords des sous-domaines et renvoient des grandeurs définies sur les interfaces. La figure 1.8 illustre le fonctionnement de ces opérateurs.

Ces opérateurs permettent d'écrire les conditions d'interface et traduisent une communication entre les sous-domaines voisins. En notant \mathbb{R}^Γ : l'espace des champs d'interface, dont la dimension est égale au rang de l'opérateur \mathbf{N} , on vérifie la propriété suivante :

$$\text{Im}\mathbf{A}^T \oplus \text{Im}\mathbf{B}^T = \mathbb{R}^\Gamma \quad (1.29)$$

C'est-à-dire qu'il est possible d'écrire une grandeur \mathbf{X} , définie sur les bords des sous-domaines, de manière unique sous la forme :

$$\mathbf{X} = \mathbf{A}^T \mathbf{X}_A + \mathbf{B}^T \mathbf{X}_B \quad (1.30)$$

avec \mathbf{X}_A et \mathbf{X}_B définies sur les interfaces.

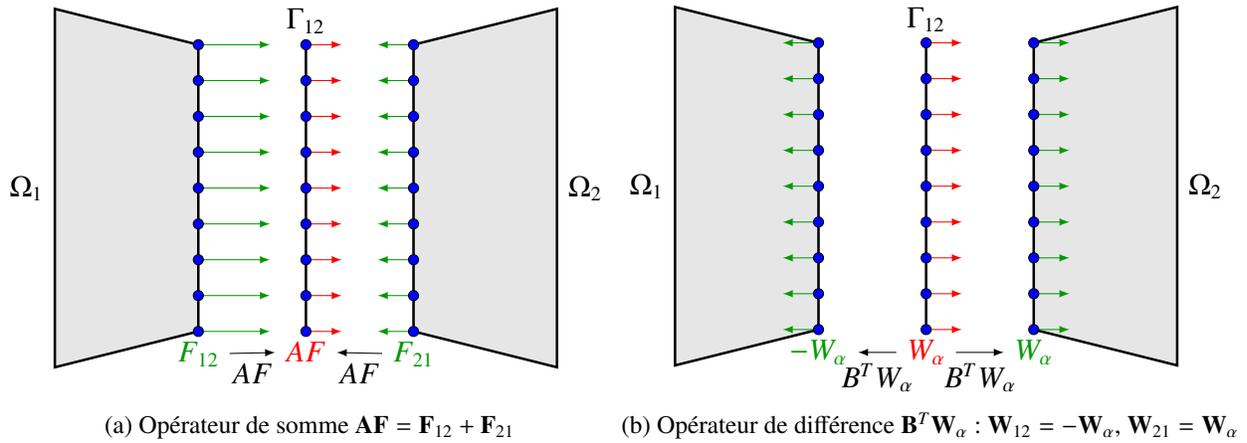


FIGURE 1.8 – Illustration des opérateurs de somme et différence

3.1.4 Problème sous-structuré

A l'aide de des notations introduites précédemment, le problème sous-structuré à résoudre devient :

$$\begin{aligned}
 \mathbf{Ku} &= \mathbf{f}_d + \mathbf{N}^T \mathbf{F} \\
 \mathbf{W} &= \mathbf{Nu} \\
 \mathbf{AF} &= 0 \\
 \mathbf{BW} &= 0
 \end{aligned} \tag{1.31}$$

3.1.5 Opérateurs et problèmes de Schur

En vue de présenter ultérieurement les approches primales et duales, il est nécessaire d'introduire auparavant le principe de condensation. Pour cela nous considérons un opérateur \mathbf{K}^s défini sur un sous-domaine Ω_s . L'opérateur condensé noté \mathbf{S}_p^s et appelé complément de Schur local primal représente le comportement de l'opérateur « vu » depuis le bord du sous-domaine Ω_s . Cet opérateur condensé est défini sur les interfaces et représente donc le comportement du sous-domaine en "boîte noire". Seuls des champs d'interface sont échangés. Un sous-domaine voisin Ω_r ne requiert pas la connaissance des champs sur l'ensemble du sous-domaine Ω_s .

Afin d'obtenir le complément de Schur de l'opérateur \mathbf{K}^s , nous considérons tout d'abord la relation d'équilibre du système 1.31 appliquée au sous-domaine Ω_s .

$$\mathbf{K}^s \mathbf{u}^s = \mathbf{f}_d + \mathbf{N}_s^T \mathbf{F}^s \tag{1.32}$$

Une renumérotation des noeuds du sous-domaine est maintenant effectuée afin de distinguer les noeuds internes (i) des noeuds de bord (b).

Avec une telle renumérotation le système devient :

$$\begin{pmatrix} \mathbf{K}_{ii}^s & \mathbf{K}_{ib}^s \\ \mathbf{K}_{bi}^s & \mathbf{K}_{bb}^s \end{pmatrix} \begin{pmatrix} \mathbf{u}_i^s \\ \mathbf{u}_b^s \end{pmatrix} = \begin{pmatrix} \mathbf{f}_d^i \\ \mathbf{f}_d^b \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{N}_b^{sT} \mathbf{F}^s \end{pmatrix} \quad (1.33)$$

Remarque 1.7

$\mathbf{N}^T \mathbf{F}$ est la grandeur physique bien définie. Cependant \mathbf{N} est de rang déficient aux coins, l'inter-effort \mathbf{F} n'y est pas proprement défini : il est défini à un inter-effort à effet nul près [Parret-Fréaud et al., 2017].

Le déplacement interne est exprimé en fonction du déplacement de bord :

$$\mathbf{u}_i^s = -\mathbf{K}_{ii}^{s-1} (\mathbf{K}_{ib}^s \mathbf{u}_b^s - \mathbf{f}_d^i) \quad (1.34)$$

Ainsi, il est possible d'exprimer le déplacement de bord en fonction seulement des efforts de bord. Cela représente le problème condensé :

$$(\mathbf{K}_{bb}^s - \mathbf{K}_{bi}^s \mathbf{K}_{ii}^{s-1} \mathbf{K}_{ib}^s) \mathbf{u}_b^s = \mathbf{S}_p^s \mathbf{u}_b^s = (\mathbf{f}_d^b - \mathbf{K}_{ii}^{s-1} \mathbf{f}_d^i) + \mathbf{N}_b^{sT} \mathbf{F}^s \quad (1.35)$$

Avec ces notations de complément de Schur local, le problème 1.31 global d'interface devient une fois condensé :

$$\begin{aligned} \mathbf{S}_p \mathbf{W} &= \mathbf{b} + \mathbf{N}_b^T \mathbf{F} \\ \mathbf{A} \mathbf{F} &= 0 \\ \mathbf{B} \mathbf{W} &= 0 \end{aligned} \quad (1.36)$$

avec \mathbf{S}_p le complément de Schur primal et \mathbf{b} les efforts imposés condensés aux interfaces.

Remarque 1.8

En écrivant le problème de manière duale et en exprimant \mathbf{F}^s en fonction de \mathbf{u}_b^s on arrive à :

$$\mathbf{S}_p^{s+} \mathbf{F}^s = \mathbf{S}_d^s \mathbf{F}^s = \mathbf{u}_b^s + \text{modes rigides} \quad (1.37)$$

avec \mathbf{S}_d^s le complément de Schur dual local qui est l'inverse généralisé du complément de Schur primal local \mathbf{S}_p^s .

Le système à résoudre comporte deux inconnues \mathbf{W} et \mathbf{F} . Il est alors possible de choisir différentes techniques de résolution. Nous nous focalisons sur les approches primales et duales. D'autres classes de méthodes telles que les méthodes à trois champs ou encore les méthodes hybrides existent pour lesquelles un bilan plus complet est fourni par [Gosselet and Rey, 2007]. Une approche mixte est présentée au chapitre 2 dans le cadre de la méthode Latin.

3.2 Approches primales

Les approches primales sont basées sur les algorithmes de types Neumann-Neumann [Bourgat et al., 1989, De Roeck and Le Tallec, 1991, Tallec et al., 1991] dont l'idée est de déterminer un déplacement continu tel que les réactions aux interfaces soient équilibrées.

3.2.1 Problème d'interface

Dans le système 1.36, les efforts sont éliminés via la condition $\mathbf{A}\mathbf{F} = 0$ ce qui conduit à :

$$\mathbf{A}\mathbf{S}_p\mathbf{W} = \mathbf{A}\mathbf{b} \quad (1.38)$$

De plus la condition de continuité des déplacements implique que le champ de déplacement est de la forme $\mathbf{W} = \mathbf{A}^T\mathbf{W}_A$ avec \mathbf{W}_A défini sur les interfaces et non pas sur les bords des sous-domaines. Ainsi \mathbf{W} est bien égal de chaque côté d'une interface. Il est alors possible de ramener le problème à l'inconnue \mathbf{W}_A .

$$(\mathbf{A}\mathbf{S}_p\mathbf{A}^T)\mathbf{W}_A = \mathbf{A}\mathbf{b} \quad (1.39)$$

3.2.2 Choix du préconditionneur

Il n'est pas question de résoudre directement le problème 1.39. Un préconditionneur astucieusement choisi doit permettre de faciliter la résolution en approximant du mieux possible l'inverse du complément de Schur global $(\mathbf{A}\mathbf{S}_p\mathbf{A}^T)^{-1}$. Qui plus est le préconditionneur doit permettre une parallélisation en minimisant le nombre d'échanges entre sous-domaines.

Le choix classique est d'approximer l'inverse du complément de Schur primal global par les contributions pondérées des inverses des compléments de Schur locaux [Klawonn and Widlund, 2001] :

$$(\mathbf{A}\mathbf{S}_p\mathbf{A}^T)^{-1} \approx \mathbf{A}^{T+}\mathbf{S}_p^+\mathbf{A}^+ \quad (1.40)$$

avec $\mathbf{A}^{T+} = \tilde{\mathbf{A}}$ un opérateur d'assemblage pondéré.

Remarque 1.9

$\mathbf{S}_p^+ = \mathbf{S}_d$ avec \mathbf{S}_d le complément de Schur dual local par sous-domaine. Calculer le pseudo-inverse revient donc à résoudre un problème de Neumann sur les sous-domaines.

3.2.3 Problème grossier

Il est à noter que la méthode BDD inclut un problème grossier afin de propager l'information à travers l'ensemble des sous-domaines accélérant ainsi la convergence de l'algorithme. Ce problème grossier est inspiré des méthodes multi-grilles [Mandel et al., 1987] et permet d'appliquer une relation d'optimalité imposant une orthogonalité entre le résidu $\mathbf{r} = \mathbf{b}_p - \underline{\mathbf{S}}_p \mathbf{W}_A$ et le noyau des compléments de Schur locaux \mathbf{R} :

$$\mathbf{R}^T \tilde{\mathbf{A}}^T \mathbf{r} = 0 \text{ ou encore } \mathbf{G}_p^T \mathbf{r} = 0 \text{ avec } \mathbf{G}_p = \tilde{\mathbf{A}} \mathbf{R} \quad (1.41)$$

Cela permet de définir le projecteur grossier \mathbf{P} :

$$\mathbf{P} = \mathbf{I} - \mathbf{G}_p \left(\mathbf{G}_p^T \underline{\mathbf{S}}_p \mathbf{G}_p \right)^{-1} \mathbf{G}_p^T \underline{\mathbf{S}}_p \quad (1.42)$$

L'algorithme classique de type BDD préconditionné avec un solveur gradient conjugué est présenté à l'algorithme 3.1.

Algorithme 3.1 : Méthode BDD préconditionnée avec un solveur gradient conjugué

Input :

$$\mathbf{P} = \mathbf{I} - \mathbf{G}_p \left(\mathbf{G}_p^T \underline{\mathbf{S}}_p \mathbf{G}_p \right)^{-1} \mathbf{G}_p^T \underline{\mathbf{S}}_p$$

$$\mathbf{W}_A^0 = \mathbf{G}_p \left(\mathbf{G}_p^T \underline{\mathbf{S}}_p \mathbf{G}_p \right)^{-1} \mathbf{G}_p^T \mathbf{b}_p$$

$$\mathbf{r}_0 = \mathbf{b}_p - \underline{\mathbf{S}}_p \mathbf{W}_A^0 = \mathbf{P}^T \mathbf{b}_p$$

$$z_0 = \tilde{\mathbf{S}}_p^{-1} \mathbf{r}_0 \text{ et } w_0 = z_0$$

for $n = 0, 1 \dots$ **do**

$$p_n = \underline{\mathbf{S}}_p \mathbf{P} w_n$$

$$\alpha_n = \frac{z_n^T r_n}{p_n^T w_n}$$

$$\mathbf{W}_A^{n+1} = \mathbf{W}_A^n + \alpha_n w_n$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n p_n$$

$$z_{n+1} = \tilde{\mathbf{S}}_p^{-1} \mathbf{r}_{n+1}$$

for $i \leq n$ **do**

$$\beta_{in} = -\frac{z_{n+1}^T p_i}{w_i^T p_i}$$

$$w_{n+1} = z_{n+1} + \sum_{i=1}^n \beta_{in} w_i$$

Il est possible d'inclure en plus du problème grossier des conditions cinématiques supplémentaires relatives aux noeuds d'interface pour le préconditionneur : par exemple imposer une continuité des déplacements pour les noeuds "coin". Cette version primale de la méthode FETI-DP discutée en 3.3.4 est la méthode BDDC [Dohrmann, 2003, Mandel and Dohrmann, 2003, Dohrmann, 2007, Mandel and Sousedík, 2007, Mandel et al., 2012].

3.2.4 Illustration de l'approche primale

Un premier problème de Dirichlet est résolu sur les sous-domaines pour imposer la continuité des déplacements aux interfaces (Figure 1.9a). Cela conduit à un déséquilibre des efforts aux interfaces (Figure 1.9b). Ce déséquilibre permet de résoudre ensuite un problème de Neumann sur les sous-domaines (Figure 1.9c). Une nouvelle discontinuité des déplacements d'interface apparaît (Figure 1.9d) qui est alors compensée par un nouveau problème de Dirichlet (Figure 1.9a).

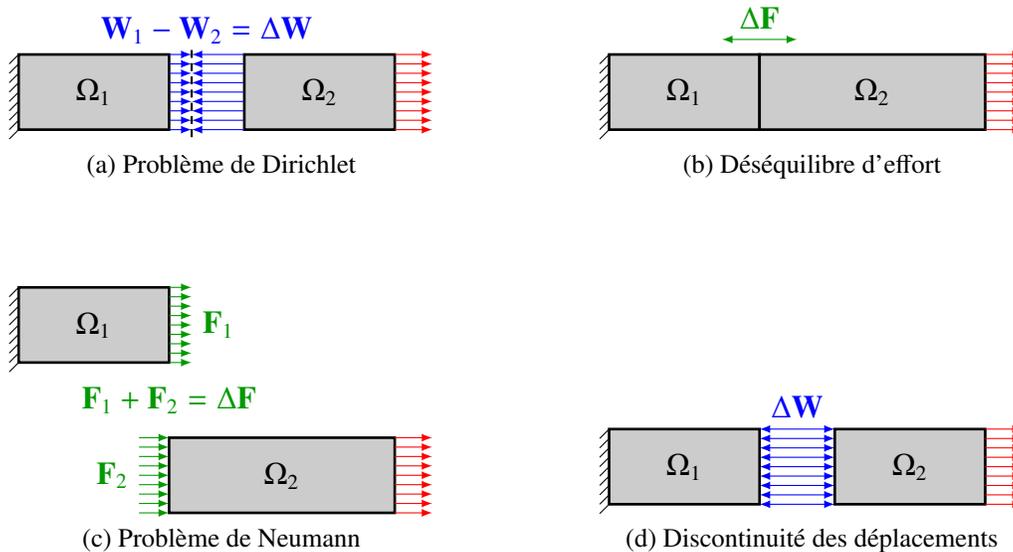


FIGURE 1.9 – Principe des algorithmes Neumann-Neumann

Ce principe des méthodes Neumann-Neumann se retrouve dans les approches primales de type BDD [Mandel, 1993, Mandel and Brezina, 1993] dont le but est d'éliminer l'inconnue duale pour obtenir un problème sur les inconnues primales. Ce problème est ensuite résolu à l'aide d'un algorithme itératif de type gradient conjugué.

3.3 Approches duales

Les approches duales de type FETI ont été introduites par Farhat et Roux [Farhat and Roux, 1991, Farhat et al., 1994] en faisant le choix d'éliminer les inconnues primales du problème sous-structuré 1.36. Ainsi l'équilibre des efforts d'interface est constamment vérifié alors que la continuité des déplacements est garantie à convergence de l'algorithme.

3.3.1 Problème d'interface

Afin de déterminer le problème d'interface écrit pour l'inconnue duale, nous exprimons d'abord \mathbf{W} en fonction de \mathbf{F} . On doit tenir compte que l'inversion du complément de Schur primal fait intervenir les mouvements de corps rigides des sous-structures ainsi qu'une condition d'orthogona-

lité entre le noyau du complément de Schur \mathbf{R} et le second membre du système. De plus comme $\mathbf{A}\mathbf{F} = 0$, il est possible de mettre \mathbf{F} sous la forme $\mathbf{F} = \mathbf{B}^T \mathbf{F}_d$. D'où le système suivant :

$$\begin{aligned} \mathbf{W} &= \mathbf{S}_p^+ (\mathbf{b} + \mathbf{B}^T \mathbf{F}_d) + \mathbf{R}\alpha \\ \mathbf{R}^T (\mathbf{b} + \mathbf{B}^T \mathbf{F}_d) &= 0 \end{aligned} \quad (1.43)$$

En remplaçant l'inverse généralisé du complément de Schur primal par le complément de Schur dual et en utilisant la condition de continuité des déplacements, le problème devient :

$$\begin{aligned} (\mathbf{B}\mathbf{S}_d\mathbf{B}^T) \mathbf{F}_d &= -\mathbf{B}\mathbf{S}_d\mathbf{b} \\ \mathbf{R}^T (\mathbf{b} + \mathbf{B}^T \mathbf{F}_d) &= 0 \end{aligned} \quad (1.44)$$

Ou encore :

$$\begin{pmatrix} \mathbf{S}_d & \mathbf{G}_d \\ \mathbf{G}_d^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{F}_d \\ \alpha \end{pmatrix} = \begin{pmatrix} -\mathbf{b}_d \\ -\mathbf{e} \end{pmatrix} \quad (1.45)$$

avec :

$$\begin{cases} \mathbf{S}_d = \mathbf{B}\mathbf{S}_d\mathbf{B}^T \\ \mathbf{b}_d = \mathbf{B}\mathbf{S}_d\mathbf{b} & \mathbf{G}_d = \mathbf{B}\mathbf{R}^T \\ \mathbf{e} = \mathbf{R}^T \mathbf{F}_d \end{cases} \quad (1.46)$$

et \mathbf{S}_d le complément de Schur dual global.

Remarque 1.10

Contrairement à l'approche primale, une contrainte d'orthogonalité doit être vérifiée afin d'assurer le caractère bien posé du problème. L'approche FETI ne peut donc pas se passer de problème grossier.

3.3.2 Choix du préconditionneur

Le préconditionneur est choisi d'une manière analogue à l'approche primale comme étant la somme pondérée des contributions locales des compléments de Schur des sous-domaines.

$$(\mathbf{B}\mathbf{S}_d\mathbf{B}^T)^{-1} \approx \mathbf{B}^{T+} \mathbf{S}_d^+ \mathbf{B}^+ \quad (1.47)$$

avec $\mathbf{B}^{T+} = \tilde{\mathbf{B}}$ un opérateur d'assemblage pondéré.

Remarque 1.11

Il est possible de chercher à approximer directement le complément de Schur primal \mathbf{S}_p présent dans le conditionneur par des évaluations de l'opérateur de rigidité \mathbf{K} sur les sous-domaines. En particulier choisir la partie \mathbf{K}_{bb} de l'opérateur relative aux noeuds d'interface permet de réduire

le temps de calcul en s'affranchissant du calcul du complément de Schur local.

3.3.3 Problème grossier

La contrainte $\mathbf{G}_d^T \mathbf{F} + \mathbf{e} = 0$ est prise en compte à l'aide d'une méthode de Krylov contrainte via l'introduction du projecteur \mathbf{P} sur $\ker \mathbf{G}_d^T$.

La solution \mathbf{F}_d est cherchée dans un espace supplémentaire telle que :

$$\begin{aligned} \mathbf{F}_d &= \mathbf{F}_{d0} + \mathbf{P}\mathbf{F}_d^* \\ \mathbf{G}_d^T \mathbf{F}_{d0} &= -\mathbf{e} \\ \mathbf{G}_d^T \mathbf{P} &= 0 \end{aligned} \quad (1.48)$$

Ainsi il est possible d'aboutir à l'algorithme 3.2.

Algorithme 3.2 : FETI préconditionnée avec un solveur gradient conjugué

Input :

$$\begin{aligned} P &= \mathbf{I} - \mathbf{G}_d (\mathbf{G}_d^T \mathbf{G}_d)^{-1} \mathbf{G}_d^T \\ \mathbf{F}_d^0 &= -\mathbf{G}_d (\mathbf{G}_d^T \mathbf{G}_d)^{-1} \mathbf{e} \\ \mathbf{r}_0 &= \mathbf{P}^T (\mathbf{b} - \mathbf{S}_d \mathbf{F}_d^0) \\ z_0 &= \mathbf{P} \tilde{\mathbf{S}}_d^{-1} \mathbf{r}_0 \text{ et } w_0 = z_0 \\ \text{for } n = 0, 1 \dots \text{ do} \\ &\left[\begin{aligned} p_n &= \mathbf{P}^T \underline{\mathbf{S}}_p w_n \\ \alpha_n &= \frac{z_n^T \mathbf{r}_n}{p_n^T w_n} \\ \mathbf{F}_d^{n+1} &= \mathbf{F}_d^n + \alpha_n w_n \\ \mathbf{r}_{n+1} &= \mathbf{r}_n - \alpha_n p_n \\ z_{n+1} &= \mathbf{P} \tilde{\mathbf{S}}_d^{-1} \mathbf{r}_{n+1} \\ \text{for } i \leq n \text{ do} \\ &\left[\beta_{in} = -\frac{z_{n+1}^T p_i}{w_i^T p_i} \right. \\ &\left. w_{n+1} = z_{n+1} + \sum_{i=1}^n \beta_{in} w_i \right] \\ \alpha &= (\mathbf{G}_d^T \mathbf{G}_d)^{-1} \mathbf{G}_d^T (-\mathbf{b} - \mathbf{S}_d \mathbf{F}_d) \\ \mathbf{u} &= \mathbf{K}^+ \mathbf{F}_d + \mathbf{R}\alpha \end{aligned} \right. \end{aligned}$$

3.3.4 Autres versions de la méthode FETI

La version présentée précédemment est appelée one-level FETI (ou FETI-1) par opposition à FETI-2 [Farhat and Mandel, 1998, Farhat et al., 1998] qui incorpore des contraintes supplémen-

taires au problème grossier pour traiter les structures plaques ou coques et des applications dynamiques. Il faut noter que la sous-structuration dynamique ne conduisait à aucun problème grossier d'où une perte d'extensibilité de la méthode. Il a donc été proposé d'ajouter un nouveau problème grossier [Farhat et al., 1995] exprimé comme une projection sur un sous-espace du problème d'interface de la FETI-1. On impose, à l'aide d'un nouveau multiplicateur de Lagrange, la continuité des déplacements transverses des noeuds "coins" des sous-domaines au cours des itérations du gradient conjugué. Un préconditionnement de Dirichlet permet une convergence optimale de la FETI-2 [Mandel et al., 1999].

Pour remédier aux limites de FETI-2 il a été proposé une version duale-primale [Farhat et al., 2001]. L'idée est similaire à FETI-2 : la continuité des déplacements aux noeuds "coins" des sous-domaines est imposée à chaque itération du gradient conjugué. Cependant, alors que cette contrainte était introduite avec un multiplicateur de Lagrange dans FETI-2, elle est désormais prise en compte directement au sein du système d'équations par une nouvelle condensation aux noeuds "coins". Ainsi dans FETI-DP il n'y a plus besoin de déterminer les mouvements de corps rigide puisque désormais les opérateurs deviennent inversibles. Cette méthode est applicable indifféremment à des cas statiques ou dynamiques avec une convergence meilleure que FETI-1 et FETI-2. De même que dans le cas de FETI-2 un préconditionnement de Dirichlet se trouve être optimal [Mandel and Tezaur, 2001]. Une version primale de FETI-DP est proposée par [Dohrmann, 2003].

Pour une implémentation plus aisée et pour retrouver certaines propriétés des mouvements de corps rigides il est proposé dans [Dostál et al., 2006] une version qui traite les conditions limites de Dirichlet par multiplicateurs de Lagrange. Cette approche est appelée total FETI ou T-FETI. Une analyse des singularités des opérateurs introduites par ces mouvements de corps rigides permet de régulariser les opérateurs de rigidité locaux. Il est montré dans [Kozubek et al., 2013] un meilleur conditionnement des opérateurs dans T-FETI que dans FETI-DP. Il y est aussi précisé les détails et manipulations pour une implémentation massivement parallèle.

Des méthodes de FETI-DP dites hybrides sont aussi proposées pour être adaptées à des calculs massivement parallèles [Klawonn and Rheinbach, 2010, Říha, 2015, Říha et al., 2017]. Une approche à trois niveaux est mise en place pour faciliter l'implémentation. Des sous-domaines connexes sont regroupés en clusters qui servent de base à la résolution parallèle inexacte du problème grossier.

La méthode FETI a aussi été adaptée pour des problèmes différents de l'élasticité linéaire. Une version de FETI-2 pour des problèmes d'acoustique est proposée par [Farhat et al., 2000a, Farhat et al., 2000b], une version de FETI-DP nommée FETI-DPH été introduite par [Farhat et al., 2005]. [Magoulès et al., 2004b] s'est intéressé à des opérateurs optimaux d'interface pour gérer des problèmes de Helmholtz. Dans le domaine des micro-systèmes électromécaniques [Confalonieri et al., 2013b] a étudié le couplage électromécanique en distinguant les deux physiques par des sous-structurations distinctes. La partie mécanique est traitée par des méthodes similaires à celles proposées par [Mahjoubi et al., 2009]. La partie électrostatique est gérée par une re-formulation de la méthode FETI. [Confalonieri et al., 2013a] s'intéresse à la décomposition de domaine pour étudier la fissuration des micro-systèmes électromécaniques de polysilicone.

4 Généralités sur le traitement du contact frottant

4.1 Formulation du problème de contact frottant

Dans le cadre de l'élasticité linéaire, la formulation variationnelle du contact a été établie par [Duvaut, 1971, Duvaut, 1976] pour lesquels une équivalence entre le problème classique de conditions de Signorini et le problème écrit sous forme d'inégalité variationnelle a été démontrée. Dans le cadre des petites perturbations sans frottement l'existence et l'unicité de la solution ont été montrées par [Kikuchi and Oden, 1988]. L'existence et l'unicité de solutions pour des problèmes de contact frottant n'ont pu être démontrées. Des exemples de solutions multiples ont été trouvés par [Ballard, 1999]. L'existence et l'unicité ne sont démontrées que pour des approches en quasi-statique et petites perturbations avec des lois de contact régularisées [Cocu et al., 1996, Hild, 2002].

Nous nous plaçons dans le cas d'un contact frottant entre un solide déformable et un solide rigide pour simplifier les notations (Figure 1.10). Les différents principes et techniques évoqués plus après sont transposables dans le cas de contact entre deux solides déformables. Un état de l'art complet peut être trouvé dans [Yastrebov, 2011]. Le problème de contact avec frottement de Coulomb peut s'écrire dans un cadre discret :

$$\begin{array}{l}
 \text{Equilibre et CL sur } \Omega : \\
 \text{Conditions normales sur } \Gamma : \\
 \text{Conditions tangentielles sur } \Gamma :
 \end{array}
 \begin{array}{l}
 \mathbf{K}\mathbf{u} = \mathbf{f}_d + \mathbf{N}^T \boldsymbol{\lambda} \\
 \left\{ \begin{array}{l} \mathbf{d}_n \geq 0 \quad \text{Non interpénétration} \\ \boldsymbol{\lambda}_n \leq 0 \quad \text{Compression seulement} \\ \mathbf{d}_n^T \boldsymbol{\lambda}_n = 0 \quad \text{Condition d'exclusion} \end{array} \right. \\
 \left\{ \begin{array}{l} \boldsymbol{\lambda}_t = 0 \\ \text{ou} \\ \text{Si } \|\boldsymbol{\lambda}_t\| < \mu |\boldsymbol{\lambda}_n| \text{ alors } \dot{\boldsymbol{\delta}}_t = 0 \\ \text{Si } \|\boldsymbol{\lambda}_t\| = \mu |\boldsymbol{\lambda}_n| \text{ alors } \exists \alpha \geq 0 \text{ tel que } \dot{\boldsymbol{\delta}}_t = -\alpha \boldsymbol{\lambda}_t \end{array} \right.
 \end{array}
 \begin{array}{l}
 \text{Contact unilatéral} \\
 \text{Adhérence} \\
 \text{Glissement}
 \end{array}
 \quad (1.49)$$

avec \mathbf{d}_n le saut de déplacement normal aux interfaces de contact qui peut s'écrire en fonction du déplacement :

$$\mathbf{d}_n = \mathbf{P}_n \mathbf{N} \mathbf{u} + \mathbf{j}_n \quad (1.50)$$

\mathbf{N} est l'opérateur qui extrait les déplacements des noeuds concernés par le contact, \mathbf{P}_n un opérateur de projection sur le solide rigide et \mathbf{j}_n le saut initial. L'effort $\boldsymbol{\lambda}$ sur la zone de contact est décomposé en effort normal de pression $\boldsymbol{\lambda}_n$ et effort tangentiel $\boldsymbol{\lambda}_t$. $\dot{\boldsymbol{\delta}}_t$ est la vitesse de glissement et μ le coefficient de frottement. Les lois de contact sont illustrées à la figure 1.11.



Remarque 1.12

Les coefficients d'adhérence et de glissement sont ici supposés égaux.

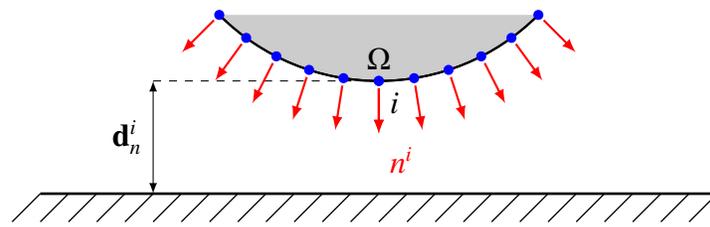


FIGURE 1.10 – Saut de déplacement et normale au contact

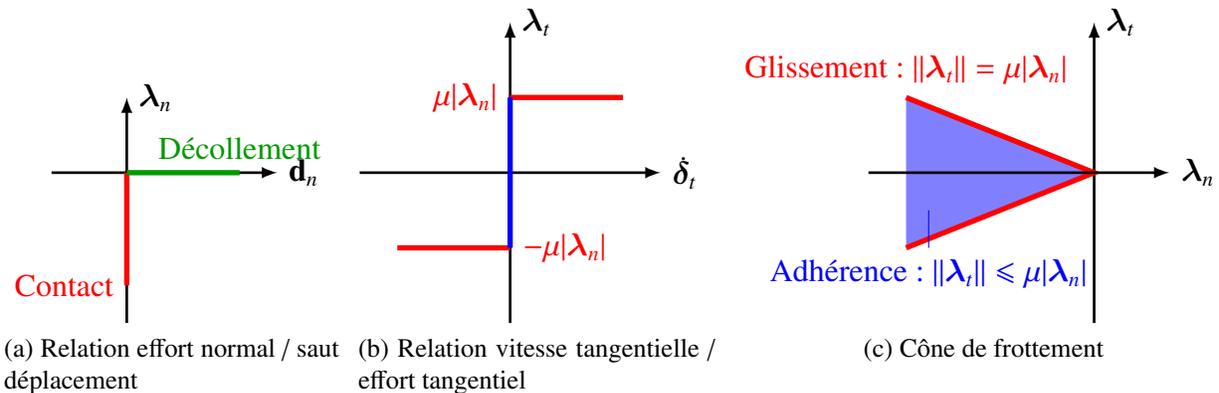


FIGURE 1.11 – Lois de contact frottant de Coulomb

La résolution de problèmes de contact avec frottement de Coulomb peut être vue comme une suite de résolutions de problèmes de contact avec frottement de Tresca dont le seuil d'adhérence/glissement ne dépend pas des efforts normaux au contact. Ce seuil est fixe et la résolution du problème est alors plus aisée.

Les problèmes de contact se gèrent à deux niveaux :

- un premier niveau qui consiste à résoudre le problème d'optimisation sous contrainte. Pour cela des méthodes itératives sont utilisées afin de prédire à chaque itération les statuts des noeuds en contact. Pour gérer le statut de contact, il est possible de citer la méthode Partial Dirichlet-Neumann ou aussi appelée méthode des statuts [Wriggers and Laursen, 2006], les algorithmes d'Uzawa [Arrow K., 1958, Bertsekas, 2014, Simo and Laursen, 1992] ou encore des algorithmes de type Newton modifié [Curnier and Alart, 1988, Alart and Curnier, 1991, Christensen, 2002, Renard, 2013]
- le second niveau consiste à convertir les informations obtenues par la boucle globale d'optimisation en conditions limites à appliquer à la structure. Les méthodes les plus utilisées sont les méthodes de pénalisation, les méthodes de lagrangien et lagrangien augmenté [Alart and Curnier, 1991]. De récentes propositions ont été faites à propos de la méthode de Nitsche [Nitsche, 1970, Wriggers and Zavarise, 2008, Juntunen and Stenberg, 2009].

Le contact sans frottement est équivalent à résoudre un problème non-linéaire d'optimisation sous des contraintes d'inégalité. Dans [Hestenes, 1969, Powell, 1969] des techniques de pénalisation et de lagrangien augmenté ont été proposées pour traiter des problèmes non-linéaires sous

contraintes d'égalité. Ces travaux ont par la suite été étendus à des problèmes d'optimisation convexes différentiables sous contrainte d'inégalité [Rockafellar, 1973] permettant ainsi de gérer les problèmes de contact sans frottement. Une extension de ces méthodes à des problèmes non différentiables pouvant ainsi traiter des cas de contact frottant peut être trouvée dans [Fortin, 1975].

4.2 Boucle interne : prise en compte des conditions limites de contact

4.2.1 Pénalisation

L'idée de la pénalisation est d'introduire au sein de l'écriture énergétique du problème les contraintes avec un coefficient de pondération. Le choix de ce coefficient représente le poids que l'on souhaite donner aux contraintes au sein de la formulation énergétique. Un coefficient élevé permet de mieux vérifier les contraintes. Il faut cependant noter qu'un trop fort poids rend le système mal conditionné et ainsi introduit des difficultés numériques dans l'inversion du problème.

Dans un cas général où l'on souhaite minimiser une fonctionnelle $\mathcal{F}(\mathbf{u})$ sous la contrainte $g(\mathbf{u})$, la méthode de pénalisation consiste à minimiser la nouvelle fonctionnelle \mathcal{H} :

$$\mathcal{H}(\mathbf{u}) = \mathcal{F}(\mathbf{u}) + \frac{1}{2}r g^2(\mathbf{u}) \quad (1.51)$$

Dans notre cas la fonctionnelle est $\mathcal{F} = \frac{1}{2}\mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{f}_d$ et la contrainte est $g = \mathbf{d}_n \geq 0$. La méthode de pénalisation associée au contact unilatéral s'écrit donc :

$$\text{Trouver } u \text{ tel que } u = \arg \min \left(\frac{1}{2}\mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{f}_d + \frac{1}{2}r \mathbf{d}_n(\mathbf{u})^T \mathbf{d}_n(\mathbf{u}) \right) \quad (1.52)$$

Une fois la minimisation effectuée le problème élément finis à résoudre est :

$$\left[\mathbf{K} + r \left(\mathbf{N}^T \mathbf{P}_n^T \mathbf{P}_n \mathbf{N} \right) \right] \mathbf{u} = \mathbf{f}_d - r \mathbf{N}^T \mathbf{P}_n^T \mathbf{j}_n \quad (1.53)$$

Remarque 1.13

Inverser ce système revient à résoudre le problème sous la contrainte d'égalité $\mathbf{d}_n = 0$ et non pas d'inégalité $\mathbf{d}_n \geq 0$. Pour cela une méthode de résolution itérative (section 4.3) doit être mise en place pour prendre en compte la non-linéarité due à la condition de non-interpénétration. La méthode de pénalisation ne permet pas de vérifier de manière exacte les contraintes.

4.2.2 Lagrangien augmenté

Contrairement aux techniques de pénalisation, l'utilisation de multiplicateurs de Lagrange permet de vérifier de manière exacte les contraintes. Cependant l'introduction de multiplicateurs transforme le problème de minimisation en une recherche de point-selle, moins facile à stabiliser. A cause de cela ils sont le plus souvent couplés à une méthode de pénalisation ce qui conduit aux techniques dites de lagrangien augmenté.

Dans le cas général introduit précédemment, l'utilisation de multiplicateur γ_n conduit au lagrangien suivant :

$$\mathcal{L}(\mathbf{u}, \gamma_n) = \mathcal{F}(\mathbf{u}) + \gamma_n g(\mathbf{u}) + \frac{1}{2} r g^2(\mathbf{u}) \quad (1.54)$$

Ainsi appliqué au contexte de problème mécanique avec contact unilatéral il est obtenu la minimisation suivante :

$$\min \left\{ \mathcal{L}(\mathbf{u}, \gamma_n) = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{f}_d + \gamma_n^T \mathbf{d}_n(\mathbf{u}) + \frac{1}{2} r \mathbf{d}_n(\mathbf{u})^T \mathbf{d}_n(\mathbf{u}) \right\} \quad (1.55)$$

La minimisation par rapport à \mathbf{u} et γ_n conduit au système :

$$\begin{cases} [\mathbf{K} + r(\mathbf{N}^T \mathbf{P}_n^T \mathbf{P}_n \mathbf{N})] \mathbf{u} + \mathbf{N}^T \mathbf{P}_n^T \gamma_n = \mathbf{f}_d \\ \mathbf{d}_n = \mathbf{P}_n \mathbf{N} \mathbf{u} = 0 \end{cases} \quad (1.56)$$

Ou encore sous forme matricielle :

$$\begin{pmatrix} \mathbf{K} + r(\mathbf{N}^T \mathbf{P}_n^T \mathbf{P}_n \mathbf{N}) & \mathbf{N}^T \mathbf{P}_n^T \\ \mathbf{P}_n \mathbf{N} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \gamma_n \end{pmatrix} = \begin{pmatrix} \mathbf{f}_d \\ 0 \end{pmatrix} \quad (1.57)$$

Remarque 1.14

La deuxième équation du système conduit bien à une vérification exacte des contraintes de contact. De la même façon que pour la pénalisation, seule une contrainte d'égalité a été imposée. Une méthode itérative pour résoudre le problème d'inégalité doit ensuite être appliquée. [Alart and Curnier, 1991] ont proposé une version légèrement différente du lagrangien augmenté pour gérer les problèmes de contact frottant et assurer une plus forte régularité de l'opérateur de contact frottant.

4.2.3 Méthode de Nitsche

La méthode de Nitsche a été introduite en 1970 [Nitsche, 1970] afin de prendre en compte des conditions limites de Dirichlet sans utiliser de multiplicateur de Lagrange. Une des forces de ces formulations de Nitsche est la grande variété des problèmes d'interface qu'il est possible de traiter [Hansbo, 2005, Juntunen and Stenberg, 2009, Embar et al., 2010].

L'autre intérêt de cette méthode est que contrairement aux techniques de lagrangien, ici aucune autre inconnue n'est rajoutée au problème. Seule l'inconnue primale est utilisée.

Pour une formulation faible classique :

$$\text{Trouver } u \in \mathcal{H}^1(\Omega) \text{ tel que : } a(u, v) = f(v) \quad \forall v \in \mathcal{H}_0^1(\Omega) \quad (1.58)$$

la prise en compte de conditions limites g de Dirichlet sur le bord Γ conduit à trouver u tel que :

$$(\nabla u, \nabla v)_\Omega - \left\langle \frac{\partial u}{\partial n}, v \right\rangle_\Gamma - \left\langle u, \frac{\partial v}{\partial n} \right\rangle_\Gamma + \sum_{E \in \Gamma} \gamma h_E^{-1} \langle u, v \rangle_E = (f, v)_\Omega - \left\langle g, \frac{\partial v}{\partial n} \right\rangle_\Gamma + \sum_{E \in \Gamma} \gamma h_E^{-1} \langle g, v \rangle_E \quad (1.59)$$

⚠ Remarque 1.15

Les termes $\sum_{E \in \Gamma} \gamma h_E^{-1} \langle u, v \rangle_E$ et $\sum_{E \in \Gamma} \gamma h_E^{-1} \langle g, v \rangle_E$ apparaissent pour stabiliser la méthode. h_E est la dimension d'un élément de bord E sur l'interface Γ . γ est le coefficient de stabilisation de la méthode et peut rappeler les techniques de pénalisation. Cependant contrairement à la pénalisation, des termes consistants sont ici ajoutés et font intervenir les dérivées normales de la variable primale u . Ces termes rappellent les techniques de lagrangien sans faire intervenir de nouvelles inconnues.

Il peut être montré que cette méthode est stable et garantit un opérateur défini positif [Becker et al., 2003]. Dans certaines applications incluant des discontinuités aux interfaces un lien peut être fait avec les méthodes de Galerkin discontinues. Cela explique un plus grand intérêt pour ces méthodes au sein de la communauté de mécanique des fluides et des couplages multi-physiques [Hansbo et al., 2004, Burman and Fernández, 2014, Johansson et al., 2016].

Si l'on applique la formulation générale 1.59 à un problème d'élasticité linéaire le problème devient :

Trouver u tel que $\forall v \in \mathcal{H}_0^1(\Omega)$:

$$\int_\Omega \mathbb{K} \epsilon(u) \epsilon(v) d\Omega - \int_\Gamma \underline{n}(\mathbb{K} \epsilon(u)) v d\Gamma - \int_\Gamma u(\mathbb{K} \epsilon(v)) \underline{n} d\Gamma + \int_\Gamma \gamma h^{-1} u v d\Gamma = \int_\Omega f v d\Omega - \int_\Gamma g(\mathbb{K} \epsilon(v)) \underline{n} d\Omega + \int_\Gamma \gamma h^{-1} g v d\Gamma \quad (1.60)$$

Cette formulation continue n'est pas transposable si aisément à une formulation discrète élément fini classique. En effet les termes d'intégration sur l'interface faisant intervenir les déformations nécessitent la détermination d'opérateurs particuliers.

Par exemple en considérant le terme $\int_\Gamma \underline{n}(\mathbb{K} \epsilon(u)) v d\Gamma$, si une discrétisation élément fini du type $v = \mathbb{N} \mathbf{V}$ est introduite on obtient :

$$\int_\Gamma \underline{n}(\mathbb{K} \epsilon(u)) v d\Gamma = \mathbf{U}^T \mathbf{P}_n^T \underbrace{\left(\int_\Gamma \mathbf{B}^T \mathbb{K} \mathbf{N} d\Gamma \right)}_{\tilde{\mathbf{K}}} \mathbf{V} \quad (1.61)$$

avec \mathbf{P}_n l'opérateur de projection suivant les normales à l'interface et $\tilde{\mathbf{K}}$ une pseudo rigidité d'interface.

Un exemple de méthode de Nitsche adaptée à un cas mécanique de contact peut être trouvé dans [Wriggers and Zavarise, 2008, Annavarapu et al., 2015].

Remarque 1.16

Bien que n'introduisant pas de nouvelle inconnue au sein de la formulation du problème, cette méthode requiert le calcul de variation d'effort de compression qui implique la détermination d'opérateur non classique. Qui plus est, les "pseudo" multiplicateurs introduits utilisent la loi de comportement matériau pour faire un lien entre une inconnue duale et l'inconnue primale de déplacement. Des limitations apparaissent inévitablement lorsque la loi de comportement se complique et ainsi un lien trivial entre les efforts de compression et le champ de déplacement n'est plus possible.

4.3 Boucle externe : algorithme d'optimisation

4.3.1 L'algorithme Partial Dirichlet-Neumann ou méthode des statuts

La méthode des statuts a pour avantage de gérer le contact très simplement. Cette méthode itérative a pour idée de remplacer les conditions de contact frottant par des conditions de Dirichlet et Neumann en fonction des statuts des nœuds. Cette méthode est itérative à deux niveaux :

- le niveau le plus bas consiste à itérer sur les nœuds d'interface et mettre à jour leur statut en fonction des calculs précédents. Par exemple si une interpénétration est détectée, une condition de Dirichlet est imposée afin d'imposer la continuité du déplacement.
- le niveau global consiste à itérer des calculs mécaniques avec les conditions de Dirichlet et Neumann mises à jour grâce au niveau précédent.

Contact normal Pour gérer les conditions normales de contact, deux statuts pour les nœuds sont définis : *contact* ou *décollement*. Pour chaque nœud d'interface un test est effectué pour vérifier si le statut est maintenu ou doit être modifié : algorithme 4.1 et Figure 1.12.

Conditions tangentielles de frottement Le cas du frottement est bien entendu géré pour les nœuds qui sont déjà définis comme étant en contact. Dans ce cas un principe similaire à la détection du contact est mis en place. Un test sur la réaction tangentielle aux nœuds est effectué : si cette réaction reste dans le cône de frottement alors les nœuds sont considérés comme adhérents, si la réaction tangentielle sort du cône alors les nœuds sont considérés comme en glissement.

Dans le cas de l'adhérence, les conditions imposées sont des conditions de Dirichlet complètes afin d'imposer la continuité du déplacement à l'interface. Le cas du glissement est légèrement plus complexe. Tout d'abord la condition de contact normale est gérée par une condition de Dirichlet partielle selon la normale à l'interface. Cela revient à imposer des conditions de type MPC [Abel and Shepard, 1979]. Les conditions tangentielles de glissement sont prises en compte via une condition partielle de Neumann qui impose une réaction en limite de cône de frottement :

$$\mathbf{F}_t^{impo} = -\mu \|\mathbf{F}_n\| \frac{\mathbf{F}_t}{\|\mathbf{F}_t\|} \quad (1.62)$$

Algorithme 4.1 : Méthode des statuts pour les conditions normales de contact

Input : Le contact est imposé sur tous les noeuds d'interface

if Décollement then

Test sur le saut de déplacement \mathbf{d}_n pour vérifier l'interpénétration :

if Interpénétration : $\mathbf{d}_n \leq 0$ then

 On change le statut du noeud en *Contact*.

 On impose la continuité du déplacement $\mathbf{d}_n = 0$ via une condition de Dirichlet.

else

 Pas de changement de statut

 La condition de Neumann nulle est maintenue

else if Contact then

Test sur la réaction normale pour vérifier les efforts de compression

if Compression : $\mathbf{F}_n \geq 0$ then

 Pas de changement de statut

 La condition de Dirichlet est maintenue

else

 Changement de statut du noeud en *Décollement*

 Relachement de la condition de continuité de Dirichlet

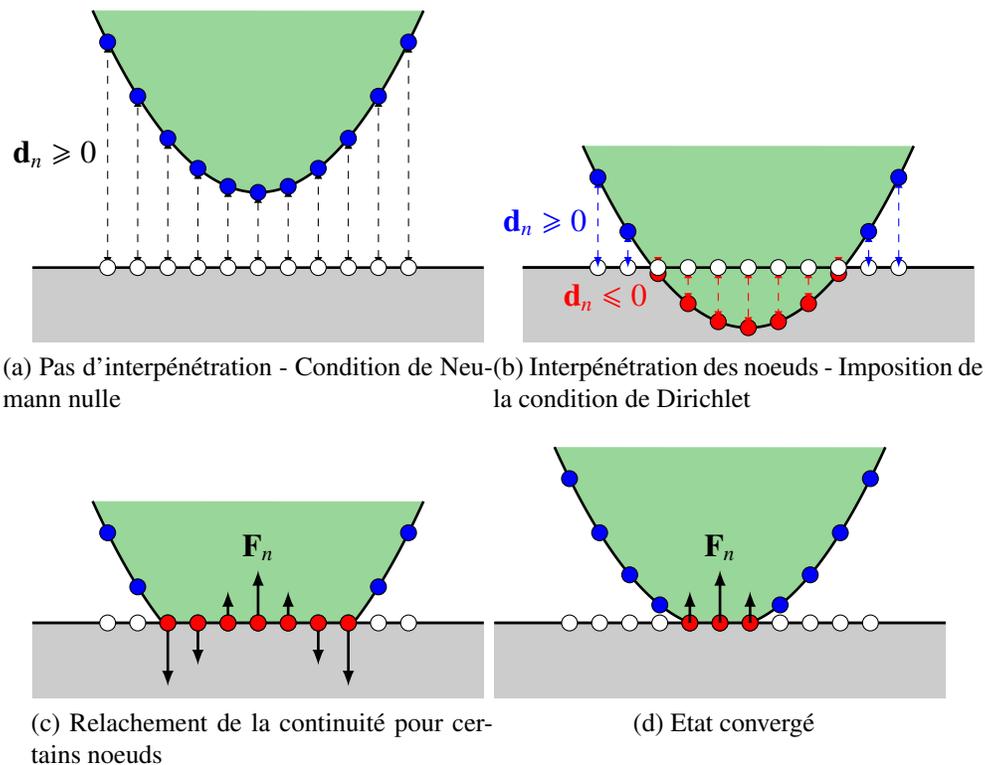


FIGURE 1.12 – Illustration de la méthode des statuts - noeuds interpénétrés ou en contact ●

⚠ Remarque 1.17

Cette méthode, simple à comprendre, a pour avantage de gérer les comportements non-linéaires et non-différentiables de contact par des conditions partielles de Dirichlet-Neumann. Cependant les conditions de Dirichlet qui changent à chaque itération nous amènent à factoriser continuellement l'opérateur de rigidité, opération coûteuse dans le cas de nombreux degrés de liberté. Les conditions de type MPC pour gérer les conditions tangentielles ne sont pas nécessairement implémentées au sein des codes industriels.

4.3.2 Algorithme de Uzawa

L'algorithme d'Uzawa est équivalent à une méthode de lagrangien avec mise à jour des multiplicateurs. Cette méthode itérative est inspirée des méthodes de return mapping [Arrow K., 1958, Giannakopoulos, 1989, Simo and Laursen, 1992] et a pour avantage de n'utiliser que des conditions limites de Neumann. Il n'y a donc pas à re-factoriser l'opérateur de rigidité à chaque itération comme c'est le cas pour la méthode des statuts. Une rigidité de type "ressort" est ajoutée aux noeuds concernés par une interpénétration. Cette rigidité a pour objectif de "ramener" les noeuds interpénétrés en contact. Ainsi lorsqu'une interpénétration est détectée, il est imposé aux noeuds concernés un effort linéairement lié à la quantité d'interpénétration. D'où le fait que plus l'interpénétration est importante, plus l'effort imposé pour lutter contre est important.

La mise à jour de l'effort imposé au contact est donc du type :

$$\mathbf{F}_{i+1} = \langle \mathbf{F}_i + k\mathbf{d}_{ni} \rangle_+ \quad (1.63)$$

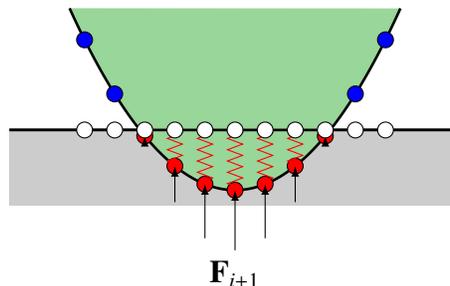


FIGURE 1.13 – Idée de la méthode d'Uzawa

Il faut faire attention au choix du coefficient de pénalisation. En effet dans le cas de contact frottant cela peut conduire à des difficultés de convergence à cause de phénomènes cycliques (par exemple cycle contact / décollement ou encore adhérence / glissement). De faibles valeurs de pondération confèrent à la méthode une plus grande robustesse. Une mise à jour de ce coefficient de pondération au cours de la résolution peut être envisagée [Kudawoo et al., 2011b, Kudawoo et al., 2011a].

Algorithme 4.2 : Algorithme d'Uzawa**Input** : Initialisation des efforts d'interface à 0**while** $\mathbf{d}_n \leq \epsilon \leq 0$ **do** Calculer : $\mathbf{F}_{i+1} = \langle \mathbf{F}_i + k\mathbf{d}_{ni} \rangle_+$

Résoudre :

 $\mathbf{K}\mathbf{u}_{i+1} = \mathbf{f}_d + \mathbf{F}_{i+1}$ Calculer : \mathbf{d}_{ni+1} **4.3.3 Méthodes de Newton**

Puisque le contact conduit à un système non linéaire, il peut sembler naturel d'utiliser un algorithme de Newton. Cependant la non-différentiabilité des lois de contact (Figure 1.11) rend complexe l'application directe de ce type d'algorithme. Des questions de convergence et stabilité se posent. Bien que non différentiables, les lois de contact peuvent être considérées comme B-différentiables. C'est-à-dire localement continues lipschitziennes et directionnellement différentiables. De plus la probabilité d'un point de vue algébrique de tomber sur la singularité est nulle ce qui permet d'appliquer des méthodes de Newton modifiées ou généralisées [Curnier and Alart, 1988, Pang, 1990, Alart and Curnier, 1991, Alart, 1997, Christensen, 2002, Renard, 2013].

⚠ Remarque 1.18

Des lois de contact régularisées rendent le problème de contact différentiable et permettent alors d'utiliser un algorithme de Newton classique. Cependant ces lois régularisées n'évitent pas les problèmes de stabilité lors des transitions adhérence / glissement.

Newton modifié En suivant les notations de [Alart and Curnier, 1991], le problème de contact peut s'écrire :

$$\mathbf{G}(\mathbf{u}) + \mathbf{H}(\mathbf{u}, \lambda) = 0 \quad (1.64)$$

avec $\mathbf{G}(\mathbf{u})$ partie différentiable et représentant l'équilibre de la structure : $\mathbf{G}(\mathbf{u}) = \mathbf{K}\mathbf{u} - \mathbf{f}_d$; et $\mathbf{H}(\mathbf{u}, \lambda)$ la partie non-différentiable de contact. La méthode de Newton modifiée propose de ré-écrire le problème précédent par :

$$\tilde{\mathbf{G}}(\mathbf{u}, \lambda) + \hat{\mathbf{H}}(\mathbf{u}, \lambda) = 0 \quad (1.65)$$

avec $\tilde{\mathbf{G}}$ différentiable et \mathbf{H} non-différentiable.

En considérant le jacobien $\tilde{\mathbf{K}} = \text{grad}\tilde{\mathbf{G}}$ comme opérateur tangent d'une méthode de Newton, l'algorithme consiste à déterminer la suite $(\mathbf{x}_i = (\mathbf{u}_i, \lambda_i))$ telle que :

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \tilde{\mathbf{K}}^{-1} (\tilde{\mathbf{G}}(\mathbf{x}_i) + \hat{\mathbf{H}}(\mathbf{x}_i)) \quad (1.66)$$

Cela conduit à l'algorithme 4.3 pour lequel $P_{C(\lambda_i^{i+1})}$ est le projecteur associé au cône de frottement.

Algorithme 4.3 : Newton modifié pour le contact frottant**Input** : Initialisation $\lambda_n^0 = 0$ et $\lambda_t^0 = 0$ **for** $i = 0 \dots m$ **do**Résoudre : $\mathbf{K}\mathbf{u}^i = \mathbf{f}_d - \mathbf{N}_n^T \lambda_n^i + \mathbf{N}_t \lambda_t^i$ Projection : $\lambda_n^{i+1} = \langle \lambda_n^i + r\mathbf{N}_n \mathbf{u}^i \rangle_-$ Projection : $\lambda_t^{i+1} = P_{C(\lambda_n^{i+1})}(\lambda_t^i + r\mathbf{N}_t \mathbf{u}^i)$  **Remarque 1.19**

L'algorithme de Newton modifié est très similaire à un algorithme d'Uzawa [Alart and Curnier, 1991].

Newton généralisé Contrairement à l'algorithme de Newton modifié pour lequel l'opérateur tangent est restreint au jacobien lié aux lois d'équilibre, ici le jacobien généralisé [Clarke, 1990, Qi and Sun, 1993] des lois de contact est pris en compte ce qui conduit à un schéma du type :

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\mathbf{K} + \mathbf{J})^{-1} (\mathbf{G}(\mathbf{x}_i) + \mathbf{H}(\mathbf{x}_i)) \quad (1.67)$$

 **Remarque 1.20**

L'expression du jacobien généralisé peut être trouvée dans [Curnier and Alart, 1988]. Les conditions de stabilité et de convergence sont présentées dans [Alart and Curnier, 1991, Alart, 1997]

5 Prise en compte du contact en décomposition de domaine

Dans le contexte des assemblages le contact intervient aux interfaces entre les différentes pièces. Les non-linéarités de contact sont donc traitées aux interfaces des sous-domaines. Les conditions d'interface ne consistent plus à imposer l'équilibre des efforts et la continuité des déplacements. Puisque il n'y a pas nécessairement continuité du déplacement, les approches primales ne sont pas adaptées aux problèmes de contact. Nous présentons en particulier dans cette section les adaptations de la méthode FETI pour le contact. Les approches mixtes permettent aussi de gérer des problèmes de contact frottant, elles seront abordées au sein des chapitres 2 et 4.

5.1 FETI pour le contact

Le problème de contact peut s'écrire comme une minimisation sous contrainte d'inégalité telle que les variables duales sont cherchées dans un sous-espace lié à la loi de contact. La minimisation sous contrainte d'inégalité se fait par une stratégie d'active-set dont l'idée est de prédire les

contraintes actives au cours des itérations. [Dureisseix and Farhat, 2001, Avery et al., 2004, Avery and Farhat, 2009] proposent des versions de FETI et FETI-DP pour le contact sans frottement nommées respectivement FETI-C et FETI-DPC. L'intégration des comportements de frottement pose de grandes difficultés. Pour gérer les problèmes de contact avec ou sans frottement, [Dostál and Horák, 2003, Dostál and Horák, 2004, Dostál et al., 2005, Dostál et al., 2009, Dostál et al., 2010, Dostál et al., 2012] ont proposé une méthodologie précise afin de gérer les contraintes d'inégalité. Cette méthode est basée sur des algorithmes d'optimisation dans le but de minimiser une fonctionnelle sous contraintes convexes [Kučera, 2008, Dostál and Kučera, 2010]. Nous présentons en particulier l'algorithme SMALBE proposé par [Dostál et al., 2003, Dostál and Horák, 2003].

5.1.1 Définition du problème et notations

Suivant les notations de [Dostál et al., 2012], le problème FETI à résoudre consiste à minimiser le lagrangien augmenté suivant :

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \rho) = \frac{1}{2} \boldsymbol{\lambda}^T (\mathbf{P}\mathbf{S}_d\mathbf{P} + \rho\mathbf{Q}) \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \mathbf{P}\mathbf{d} + \boldsymbol{\mu}^T \mathbf{G}\boldsymbol{\lambda} \quad (1.68)$$

avec \mathbf{P} et \mathbf{Q} les projecteurs orthogonaux respectivement sur le noyau de \mathbf{G} et l'image de \mathbf{G}^T , \mathbf{d} lié aux conditions limites. Nous notons \mathbf{g} le gradient du lagrangien par rapport à $\boldsymbol{\lambda}$:

$$\mathbf{g}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \rho) = (\mathbf{P}\mathbf{S}_d\mathbf{P} + \rho\mathbf{Q}) \boldsymbol{\lambda} - \mathbf{P}\mathbf{d} + \mathbf{G}^T \boldsymbol{\mu} \quad (1.69)$$

Remarque 1.21

Le problème consiste à minimiser \mathcal{L} sous la contrainte d'admissibilité de $\boldsymbol{\lambda} \in \Lambda$, avec :

$$\Lambda = \{\boldsymbol{\lambda}, \lambda_n \leq 0 \text{ et } \|\boldsymbol{\lambda}_i\| \leq \mu|\lambda_n|\} \quad (1.70)$$

S'il est considéré $\mathcal{N} = \{1, 2, \dots, n\}$ avec n le nombre de degrés de liberté du problème, il est possible de construire les ensembles : \mathcal{I} l'ensemble des indices des variables contraintes, \mathcal{E} l'ensemble des variables non contraintes, \mathcal{A} un *active set* de $\boldsymbol{\lambda}$ et \mathcal{F} un *free set* de $\boldsymbol{\lambda}$.

$$\begin{aligned} \mathcal{A}(\boldsymbol{\lambda}) &= \{i \in \mathcal{I}, \lambda_i = 0\} \\ \mathcal{F}(\boldsymbol{\lambda}) &= \{i \in \mathcal{E}, \text{ ou } \lambda_i < 0\} \end{aligned} \quad (1.71)$$

Nous définissons maintenant le gradient projeté \mathbf{g}^P tel que

$$\mathbf{g}^P(\boldsymbol{\lambda}) = \phi(\boldsymbol{\lambda}) + \beta(\boldsymbol{\lambda}) \quad (1.72)$$

avec ϕ et β respectivement le *free gradient* et le *chopped gradient* définis par :

$$\phi_i(\boldsymbol{\lambda}) = \begin{cases} \mathbf{g}_i(\boldsymbol{\lambda}) & \text{pour } i \in \mathcal{F}(\boldsymbol{\lambda}) \\ 0 & \text{pour } i \in \mathcal{A}(\boldsymbol{\lambda}) \end{cases} \quad \beta_i(\boldsymbol{\lambda}) = \begin{cases} 0 & \text{pour } i \in \mathcal{F}(\boldsymbol{\lambda}) \\ \mathbf{g}_i^-(\boldsymbol{\lambda}) & \text{pour } i \in \mathcal{A}(\boldsymbol{\lambda}) \end{cases} \quad (1.73)$$

Le problème est résolu grâce à un algorithme d'optimisation à deux niveaux. Une boucle externe permet l'approximation des multiplicateurs liés aux contraintes d'égalité alors que la minimisation due aux contraintes d'inégalité est traitée au sein de la boucle interne. Nous présentons premièrement la boucle externe qui correspond à l'algorithme SMALBE (*Semi Monotonic Augmented Lagrangian for Bound and Equality constrained QP problems*) proposé par [Conn et al., 1991, Dostál and Horák, 2003]. Dans un second temps est détaillé l'algorithme interne de minimisation sous contrainte d'inégalité MPRGP (*Modified Proportioning with Reduced Gradient Projection*) [Dostál and Schoberl, 2002].

5.1.2 SMALBE : Semi Monotonic Augmented Lagrangian for Bound and Equality constrained QP problems

De nombreuses applications et résultats de convergence peuvent être trouvées dans [Dostál and Horák, 2003, Dostál, 2006a, Dostál, 2006b] en ce qui concerne la première version de l'algorithme 5.1.

Algorithme 5.1 : SMALBE

```

Input :  $\alpha \in ]0, 1[$ ,  $\beta > 1$ ,  $\rho_0 > 0$ ,  $\eta_0 > 0$ ,  $M > 0$ ,  $\mu_0$ 
for  $k=0, \dots, m$  do
    Trouver  $\lambda_k \in \Lambda$  tel que  $\|\mathbf{g}^P(\lambda_k, \mu_k, \rho_k)\| \leq M\|\mathbf{G}\lambda_k\|$ 
    if  $\|\mathbf{g}^P(\lambda_k, \mu_k, \rho_k)\|$  et  $\|\mathbf{G}\lambda_k\|$  sont assez petits then
        stop
    else
        Mise à jour des multiplicateurs :
         $\mu_{k+1} = \mu_k + \rho_k \mathbf{G}\lambda_k$ 
        Mise à jour de la pénalisation :
        if  $\|\mathbf{G}\lambda_k\| \leq \eta_k$  then
             $\rho_{k+1} = \rho_k$ 
             $\eta_{k+1} = \alpha\eta_k$ 
        else
             $\rho_{k+1} = \beta\rho_k$ 
             $\eta_{k+1} = \eta_k$ 

```

Remarque 1.22

La première étape de l'algorithme qui consiste à effectuer une minimisation d'une fonction convexe quadratique sous contrainte d'inégalité est résolue par un algorithme spécifique décrit à la partie 5.1.3.

Une version de l'algorithme avec mise à jour du coefficient M (Algorithme 5.2) sans mettre à jour la pénalisation est présentée dans [Dostál, 2009, Dostál and Kučera, 2010]. Dans ce cas le gradient projeté est modifié en K-gradient $\mathbf{g}^K = \phi + \beta$ avec ϕ le *free gradient* et β le *K-boundary gradient* définis par :

$$\phi_i(\boldsymbol{\lambda}) = \begin{cases} \mathbf{g}_i(\boldsymbol{\lambda}) & \text{pour } i \in \mathcal{F}(\boldsymbol{\lambda}) \\ 0 & \text{pour } i \in \mathcal{A}(\boldsymbol{\lambda}) \end{cases} \quad \beta_i(\boldsymbol{\lambda}) = \begin{cases} 0 & \text{pour } i \in \mathcal{F}(\boldsymbol{\lambda}) \\ \mathbf{g}_i + \frac{\|\mathbf{g}_i\|}{\|\nabla \lambda_i\|} \nabla \lambda_i & \text{pour } i \in \mathcal{A}(\boldsymbol{\lambda}) \end{cases} \quad (1.74)$$

Algorithme 5.2 : SMALBE-M

Input : $\alpha \in]0, 1[$, $\beta > 1$, $\rho > 0$, $M_0 > 0$, $\boldsymbol{\mu}_0$

for $k=0, \dots, m$ **do**

Trouver $\boldsymbol{\lambda}_k \in \Lambda$ tel que $\|\mathbf{g}^P(\boldsymbol{\lambda}_k, \boldsymbol{\mu}_k, \rho_k)\| \leq M\|\mathbf{G}\boldsymbol{\lambda}_k\|$

if $\|\mathbf{g}^P(\boldsymbol{\lambda}_k, \boldsymbol{\mu}_k, \rho_k)\|$ et $\|\mathbf{G}\boldsymbol{\lambda}_k\|$ sont assez petits **then**

└ stop

else

Mise à jour des multiplicateurs : $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \rho_k \mathbf{G}\boldsymbol{\lambda}$

Mise à jour du coefficient M :

if $k > 0$ et $\mathcal{L}(\boldsymbol{\lambda}_k, \boldsymbol{\mu}_k, \rho) < \mathcal{L}(\boldsymbol{\lambda}_{k-1}, \boldsymbol{\mu}_{k-1}, \rho) + \frac{\rho}{2}\|\mathbf{G}\boldsymbol{\lambda}_k\|^2$ **then**

└ $M_{k+1} = \frac{M_k}{\beta}$

else

└ $M_{k+1} = M_k$

Une version pour prendre en compte une plus grande variété de contraintes est SMALSE-M pour *Semi Monotonic Augmented Lagrangian for Separable and Equality constraints* [Dostál and Kučera, 2010, Dostál et al., 2012]. Il s'agit d'introduire au sein du problème de minimisation sous contrainte un paramètre de contrôle η tel que cela devienne :

$$\text{Trouver } \boldsymbol{\lambda}_k \in \Lambda \text{ tel que } \|\mathbf{g}^P(\boldsymbol{\lambda}_k, \boldsymbol{\mu}_k, \rho_k)\| \leq \min(M\|\mathbf{G}\boldsymbol{\lambda}_k\|, \eta) \quad (1.75)$$

5.1.3 Minimisation sous contrainte

L'algorithme de minimisation est dérivé des travaux de [Polyak, 1969] qui proposait un algorithme de résolution de problème quadratique contraint par des bornes simples sur le domaine de définition : par exemple en 1D $\Omega_S = [0, \infty]$. Cette méthode était basée sur une *restarted conjugate gradient* qui permettait à chaque itération de définir un sous-ensemble de variables contraintes sur les bords du domaine Ω_S . Un gradient conjugué était ensuite appliqué pour minimiser la fonctionnelle voulue. Des améliorations à cet algorithme ont été proposées afin de prendre en compte des contraintes plus complexes. En particulier l'ajout de projections du gradient a été introduite par [More and Toraldo, 1991]. La résolution d'un problème inexact dans la boucle interne permet un

gain en temps de calcul et cela doit être couplé à un contrôle adaptatif de la précision du calcul [Friedlander and Martinez, 1994, Dostál, 1997]. Nous présentons le principe de l'algorithme dans un cas général de minimisation sous contrainte de non négativité [Dostál and Schoberl, 2002]. Une version dans le cadre de contraintes plus générales peut être trouvée dans [Kučera, 2008, Dostál and Kučera, 2010].

$$\text{minimiser } q(\boldsymbol{\lambda}) = \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{A} \boldsymbol{\lambda} - \mathbf{b} \text{ tel que } \boldsymbol{\lambda} \in \mathcal{I} = \{\boldsymbol{\lambda}, \boldsymbol{\lambda} \geq 0\} \quad (1.76)$$

De la même façon que précédemment il est défini un gradient projeté \mathbf{g}^P à partir du *free gradient* ϕ et du chopped gradient β . L'algorithme est basé sur trois étapes :

1. *Expansion step* :

$$\boldsymbol{\lambda}_{k+1} = \mathbf{P}_{\mathcal{I}} \left(\boldsymbol{\lambda}_k - \underline{\alpha} \phi(\boldsymbol{\lambda}_k) \right) \quad (1.77)$$

avec $\underline{\alpha} \in (0, \|A\|^{-1}]$ le pas de l'expansion, $\mathbf{P}_{\mathcal{I}}$ le projecteur sur le domaine Ω_S . L'étape d'expansion a pour objectif d'étendre le domaine des variables contraintes. En introduisant le *reduced free gradient* $\tilde{\phi}_i = \min(\boldsymbol{\lambda}_i / \underline{\alpha}, \phi_i)$, l'itéré $\boldsymbol{\lambda}_k$ est dit strictement proportionnel pour un Γ donné s'il vérifie :

$$\|\beta(\boldsymbol{\lambda}_k)\|^2 \leq \Gamma^2 \tilde{\phi}(\boldsymbol{\lambda}_k)^T \phi(\boldsymbol{\lambda}_k) \quad (1.78)$$

Le principe de cette étape d'expansion est illustré à la Figure 1.14a.

2. *Proportioning step* :

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \alpha_{cg} \phi(\boldsymbol{\lambda}_k) \quad (1.79)$$

avec α_{cg} qui minimise $q(\boldsymbol{\lambda}_k - \alpha \phi(\boldsymbol{\lambda}_k))$:

$$\alpha_{cg} = \frac{\phi(\boldsymbol{\lambda}_k)^T \mathbf{g}(\boldsymbol{\lambda}_k)}{\phi(\boldsymbol{\lambda}_k)^T \mathbf{A} \phi(\boldsymbol{\lambda}_k)^T} \quad (1.80)$$

Cette étape de proportioning a pour objectif de réduire le champ des variables contraintes. Le principe est illustré à la Figure 1.14b.

3. *Conjugate gradient step* :

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \alpha_{cg} \mathbf{p}_k \quad (1.81)$$

avec \mathbf{p} la direction du gradient conjugué déterminée par :

$$\mathbf{p}_{k+1} = \phi(\boldsymbol{\lambda}_{k+1}) - \gamma \mathbf{p}_k \quad (1.82)$$

$$\text{avec } \gamma = \frac{\phi(\boldsymbol{\lambda}_{k+1})^T \mathbf{A} \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}$$

Avec ces trois étapes il est possible de construire l'algorithme MPRGP (ALG. 5.3).

Algorithme 5.3 : MPRGP : Modified Proportioning with Reduced Gradient Projection

Input : $\lambda_0 \in \mathcal{I}$, $\underline{\alpha} \in (0, \|A\|^{-1}]$, $\Gamma > 0$

for $k = 1, \dots, m$ **do**

- if** $\mathbf{g}^P(\lambda_k) = 0$ **then**
 - └ $\lambda_{k+1} = \lambda_k$
- else if** λ_k est strictement proportionnel et $\mathbf{g}^P(\lambda_k) \neq 0$ **then**
 - └ Etape de GC pour générer $\lambda_{k+1/2}$
 - └ **if** $\lambda_{k+1/2} \in \mathcal{I}$ **then**
 - └ $\lambda_{k+1} = \lambda_{k+1/2}$
 - └ **else**
 - └ Etape d'expansion pour générer λ_{k+1} à partir de $\lambda_{k+1/2}$
- else if** λ_k n'est pas strictement proportionnel **then**
 - └ Etape de proportioning pour générer λ_{k+1}

Remarque 1.23

Un version appelée KPRGP utilise le K-gradient à la place du gradient projeté [Kučera, 2008]. Une description avec une vision plus algébrique est présentée à l'algorithme 5.4 tiré de [Dostál and Schoberl, 2002].

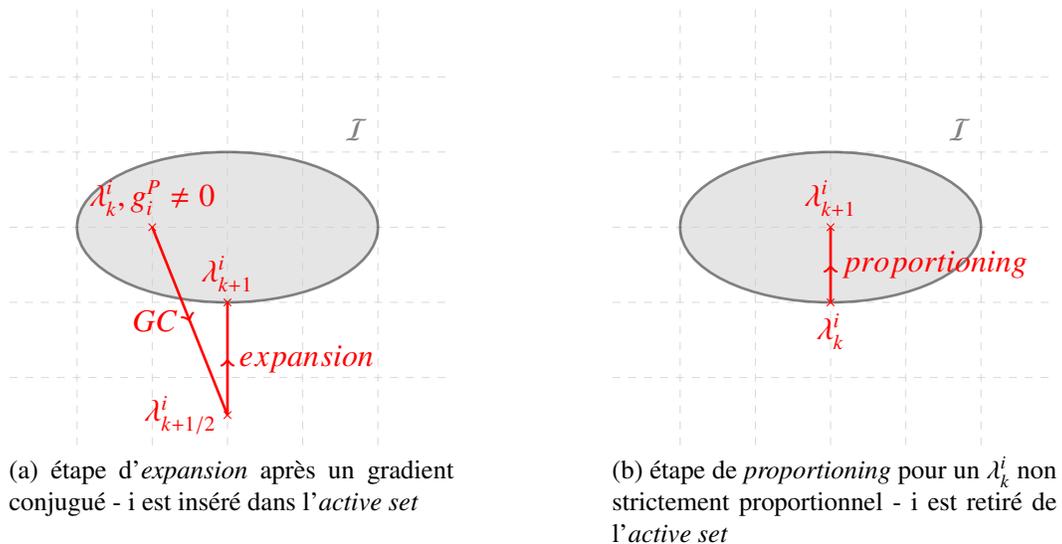


FIGURE 1.14 – Illustration des différentes étapes

Algorithme 5.4 : MPRGP : Modified Proportioning with Reduced Gradient Projection

Input : $\lambda_0 \in \Omega_S$, $\underline{\alpha} \in (0, \|A\|^{-1}]$, $\Gamma > 0$, $k = 0$, $r = A\lambda_0 - \mathbf{b}$, $\mathbf{p}_0 = \phi(\lambda_0)$

while $\mathbf{g}^P(\lambda_k) > \epsilon$ **do**

if $\|\beta(\lambda_k)\|^2 \leq \Gamma^2 \tilde{\phi}(\lambda_k)^T \phi(\lambda_k)$ **then**

 Etape de proportioning et test du gradient conjugué :

$$\alpha_{cg} = \frac{r^T \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}, \quad y = \lambda_k - \alpha_{cg} \mathbf{p}_k$$

$\alpha_f = \max(\alpha, \lambda_k - \alpha \mathbf{p}_k \in \mathcal{I})$ **if** $\alpha_{cg} \leq \alpha_f$ **then**

 Etape de gradient conjugué :

$$\lambda_{k+1} = y, \quad r = r - \alpha_{cg} A \mathbf{p}_k$$

$$\gamma = \frac{\phi(y)^T A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}, \quad \mathbf{p}_{k+1} = \phi(y) - \gamma \mathbf{p}_k$$

else

 Etape d'expansion :

$$\lambda_{k+1/2} = \lambda_k - \alpha_f \mathbf{p}_k, \quad r = r - \alpha_f A \mathbf{p}_k$$

$$\lambda_{k+1} = \mathbf{P}_{\mathcal{I}}(\lambda_{k+1/2} - \underline{\alpha} \phi(\lambda_{k+1/2}))$$

$$r = A\lambda_{k+1} - \mathbf{b}, \quad \mathbf{p}_{k+1} = \phi(\lambda_{k+1})$$

else

 Etape de proportioning :

$$d = \beta(\lambda_k), \quad \alpha_{cg} = \frac{r^T d}{d^T A d}$$

$$\lambda_{k+1} = \lambda_k - \alpha_{cg} d, \quad r = r - \alpha_{cg} A d, \quad \mathbf{p}_{k+1} = \phi(\lambda_{k+1})$$

$k = k + 1$

5.2 Méthodes de Schwarz pour le contact

Peu de travaux concernent la gestion des problèmes de contact avec des méthodes de Schwarz. Les méthodes avec recouvrement sont intrinsèquement peu adaptées aux discontinuités de déplacement dues au contact entre sous-domaines. Les travaux de Badea [Badea and Wang, 1999, Badea et al., 2003] présentent des méthodes de Schwarz pour prendre en compte des problèmes d'inégalité variationnelle. Les méthodes de Schwarz sont alors vues comme des algorithmes de correction de sous-espace (*subspace correction algorithm*) Le contact est introduit dans ces méthodes par [Badea and Krause, 2012, Badea and Lebon, 2015]. Cependant les lois de contact ne sont pas appliquées aux interfaces entre les sous-domaines. Le contact n'est géré qu'en "conditions limites" du domaine global. Nous pouvons aussi regretter dans ces papiers qu'aucun exemple numérique n'ait été présenté pour illustrer les performances de l'algorithme.

6 Bilan

Nous avons présenté au sein de ce chapitre les principales méthodes de décomposition de domaine : les [méthodes de Schwarz](#) avec ou sans recouvrement [Lions, 1988, Lions, 1990, Dryja

and Widlund, 1994, Cai and Sarkis, 1999], les approches primales [Tallec et al., 1991] à travers la méthode BDD [Mandel, 1993] ou encore les approches duales avec la méthode FETI [Farhat and Roux, 1991, Farhat et al., 2001]. Une approche mixte sera présentée au chapitre 2 dans le cadre de la méthode Latin [Ladevèze, 1999]. Ces méthodes ont été présentées pour le cas de l'élasticité linéaire sous l'hypothèse des petites perturbations.

Le problème de contact frottant de Coulomb a été introduit à la section 4. Nous avons présenté une description des algorithmes de traitement pour ce type de non-linéarité. La méthodes des statuts, d'Uzawa [Arrow K., 1958, Simo and Laursen, 1992] ou encore des algorithmes de type Newton [Alart and Curnier, 1991, Alart, 1997] permettent de gérer la contrainte d'inégalité introduite par les lois de contact. Ces méthodes informent sur les contraintes à appliquer aux noeuds d'interface via des techniques de pénalisation ou lagrangien augmenté. Nous avons aussi choisi de présenter dans un cadre plus général la méthode de Nitsche [Nitsche, 1970] qui permet de prendre en compte des conditions limites de type Dirichlet de manière exacte sans introduire de multiplicateurs. Une application à un cas de contact est proposée par [Wriggers and Zavarise, 2008].

Une dernière section a permis de dresser un état de l'art du traitement du contact au sein des méthodes de décomposition de domaine. [Dureisseix and Farhat, 2001, Avery and Farhat, 2009] ont proposé une méthodologie pour gérer le contact sans frottement au sein de FETI et FETI-DP. Cependant les avancées et travaux récents utilisent les algorithmes proposés par Dostál [Dostál and Horák, 2003, Dostál et al., 2005] grâce à un cadre mathématique rigoureux d'optimisation de fonctionnelle sous contrainte d'inégalité [Kučera, 2008, Dostál, 2009, Dostál and Kučera, 2010]. Cela permet en particulier de traiter des problèmes de contact frottant de type Coulomb avec une approche de type FETI. Les méthodes de Schwarz avec recouvrement, ainsi que les approches primales ne peuvent être adaptées pour gérer le contact au niveau des interfaces entre sous-domaines.

Les tentatives d'intégration de ces différentes méthodes dans un code industriel n'ont jamais débouché sur de francs succès. Dans le contexte de l'élasticité linéaire en petites perturbations les approches primales ou duales pourraient utiliser des fonctionnalités déjà présentes au sein des codes industriels : résoudre des problèmes éléments finis sur des sous-domaines, extraire des champs de bords... Cependant les méthodes spécifiques pour résoudre des problèmes non-linéaires ne sont pas aussi facilement implémentables dans les codes éléments finis et demeurent extrêmement intrusives. L'idée générale de la non-intrusivité est d'éviter tout développement interne au sein des codes industriels et toute modification de la mise en donnée d'un problème mécanique. Nous pouvons citer des travaux sur la méthode *Globale/Locale* [Gendre et al., 2009, Guguin et al., 2014, Guguin et al., 2016] où une technique de *sub-modelling* exacte est mise en place requérant uniquement des analyses éléments finis implémentées dans les codes industriels. Ces techniques de *sub-modelling* permettent de traiter des phénomènes locaux non-linéaires variés : de la plasticité [Gendre et al., 2009, Blanchard et al., 2017], de la propagation de fissuration [Passieux et al., 2013, Duval et al., 2016]. D'autres travaux en dynamique explicite sont proposés par [Bettinotti et al., 2014]. Les approches mixtes que nous aborderons au chapitre 2 et qui proposent une prise en compte du contact efficace [Champaney, 1996, Blanžé et al., 2000] font intervenir des opérateurs d'interface complexes et ne peuvent non plus être facilement implémentées dans les codes industriels.

Implémentation non-intrusive de la méthode Latin

Au sein de ce chapitre est présentée la méthode Latin dans le cadre de la décomposition de domaine. La première section introduit les principes généraux de la méthode Latin comme solveur non-linéaire.

La deuxième section se focalise sur son application à la décomposition de domaine et à ses aspects intrusifs pour les codes industriels. Cela conduit à proposer dans la troisième section une implémentation non-intrusive de la méthode Latin en décomposition de domaine, illustrée par quelques exemples de validation.

Sommaire

1	Introduction - la méthode Latin : un solveur non-linéaire	47
1.1	Principes généraux de la méthode Latin	47
1.2	Quelques cas d'application	49
1.3	Méthode Latin multi-paramétrique	49
2	Méthode Latin appliquée à la décomposition de domaine mono-échelle	50
2.1	Obtention du problème	50
2.2	Convergence	52
2.3	Interprétation des directions de recherches	52
2.4	Etape linéaire intrusive	53

3	Vision non-intrusive par direction de recherche non-locale	54
3.1	Direction de recherche non-locale	54
3.1.1	Principe de la direction de recherche non intrusive	54
3.1.2	Direction de recherche non-intrusive via un problème de Dirichlet	55
3.1.3	Etape linéaire non-intrusive	56
3.1.4	Etape locale non-locale	57
3.1.5	Prise en compte des conditions limites	59
3.2	Validation sur un cas simple 1D	59
3.2.1	Comparaison avec une implémentation classique	59
3.2.2	Influence des directions de recherche	60
3.3	Présentation d'une simulation 3D	62
4	Bilan	67

1 Introduction - la méthode Latin : un solveur non-linéaire

1.1 Principes généraux de la méthode Latin

La méthode Latin, pour *Large Time INcremental method*, est proposée par [Ladevèze, 1985, Ladevèze, 1999] pour résoudre des problèmes non-linéaires dont la formulation permet une distinction des équations locales et des équations linéaires. La méthode Latin est basée sur trois principes :

- P1** : une séparation des équations en deux groupes : l'un constitué des équations locales et éventuellement non-linéaires, l'autre rassemblant les équations linéaires. Cela permet de définir deux groupes de solutions partielles du problème. L'espace *linéaire ou global* \mathcal{A} regroupe les solutions des équations linéaires, l'espace *local* \mathcal{L} regroupe les solutions des équations non-linéaires.
- P2** : un schéma itératif pour déterminer successivement des solutions partielles sur chaque espace afin de converger itérativement vers l'intersection des deux espaces : la solution du problème complet. Ce schéma itératif est caractérisé par deux directions de recherche. L'une k^+ pour passer de \mathcal{A} à \mathcal{L} , l'autre k^- de \mathcal{L} à \mathcal{A} .
- P3** : une représentation adaptée des inconnues, initialement décomposition radiale puis PGD temps-espace.

Remarque 2.0

Nous n'introduisons pas ici de représentation des inconnues sous la forme d'une décomposition temps-espace. Nous utiliserons le fait que les différents opérateurs sont constants et indépendants du temps permettant une factorisation en début d'algorithme et ensuite une résolution peu coûteuse au cours des itérations. Le principe de la méthode Latin est illustré à la figure 2.1.

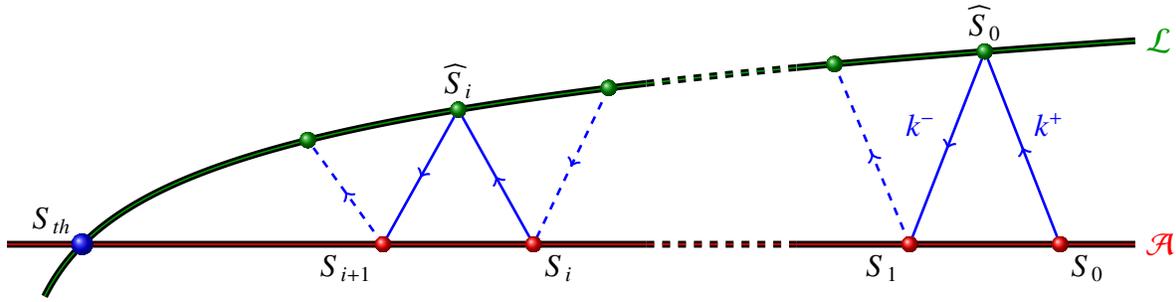


FIGURE 2.1 – Principe de la méthode Latin

La méthode Latin se caractérise par l'enchaînement successif de deux étapes :

1. une *étape locale* éventuellement non-linéaire qui consiste à déterminer une solution \widehat{S}_i de \mathcal{L} connaissant une solution S_i de \mathcal{A} suivant la direction de recherche k^+
2. une *étape linéaire* éventuellement globale qui consiste à déterminer une solution S_{i+1} de \mathcal{A} connaissant une solution \widehat{S}_i de \mathcal{L} suivant la direction de recherche k^-

⚠ Remarque 2.1

- En reprenant l'illustration de la méthode Latin, les directions de recherche k^+ et k^- sont aussi appelées respectivement direction de recherche de *montée* et direction de recherche de *descente*.
- Dans un cadre général, les solutions S_i s'écrivent de la forme $S_i = [\dot{\epsilon}_p, \dot{\mathbb{X}}, \sigma, \mathbb{Y}]$, avec $\dot{\epsilon}_p$ le taux de déformation plastique, $\dot{\mathbb{X}}$ regroupant les autres variables internes, et leurs grandeurs duales respectives σ et \mathbb{Y} .

Le caractère non-incremental de la méthode Latin est aussi un élément qui la distingue des autres techniques de résolution de problèmes non-linéaires. Contrairement à un algorithme de type Newton, ici à chaque itération une approximation de la solution pour chaque pas de temps / incrément est déterminée. Cela conduit à l'algorithme 1.1 résumant la méthode Latin.

Algorithme 1.1 : Méthode Latin

Input : Initialisation à $S_0 \in \mathcal{A}$

while *Non convergence* **do**

Étape locale :

 Trouver pour chaque pas de temps $\widehat{S}_i \in \mathcal{L} \cap k^+$ connaissant une solution $S_i \in \mathcal{A}$

Étape linéaire :

 Trouver pour chaque pas de temps $S_{i+1} \in \mathcal{A} \cap k^-$ connaissant une solution $\widehat{S}_i \in \mathcal{L}$

Relaxation :

$S_{i+1} \leftarrow \mu S_{i+1} + (1 - \mu) S_i$

 **Remarque 2.2**

A la fin de chaque étape linéaire il est ajouté une étape de relaxation pour garantir la convergence de la méthode. Cependant cette technique de relaxation ne fait qu'intervenir deux itérés successifs. Nous proposons au chapitre 3 (section 5) une nouvelle vision de la relaxation pour le cas d'interfaces parfaites.

1.2 Quelques cas d'application

Les premières non-linéarités gérées par la méthode à grand incrément de temps sont les problèmes non-linéaires matériaux. Le cas de l'élasto-plasticité est abordé en 1987 [Boisse, 1987], la visco-plasticité en 1989 [Cognard, 1989], les phénomènes cycliques par [Arzt, 1994]. Des applications dans le cadre de la thermo-mécanique sont proposées par [Cognard et al., 1999]. Les non-linéarités géométriques dues aux grands déplacements ont été étudiées dès 1991 [Vauchez, 1991, Boucard, 1996]. Les phénomènes d'endommagement de composites introduits par [Allix and Ladevèze, 1992] ont été couplés avec du non-linéaire géométrique par [Saavedra Redlich, 2012]. De l'endommagement pour des structures en béton armé est traité dans [Vitse, 2017]

Le principe de séparer les équations permet aussi d'appliquer la méthode Latin non pas nécessairement à des problèmes non-linéaires mais à des problèmes dont on souhaiterait découpler certains comportements. Les problèmes multi-physiques [Dureisseix et al., 2003, Néron, 2007] sont alors traités en distinguant les physiques l'une de l'autre afin de simplifier la résolution du problème. L'application de la méthode Latin à la décomposition de domaine suit la même idée : on considère de façon séparée les sous-domaines des interfaces. Les premiers travaux dans ce cadre là ont été proposés par [Lorong, 1994] puis adaptés aux architectures informatiques parallèles par [Dureisseix, 1997]. Cette distinction sous-domaine/interface permet d'introduire aisément des comportements particuliers aux interfaces comme les inter-plis pour les composites [Kerfriden et al., 2009, Saavedra Redlich, 2012] ou encore des comportements de contact pour des assemblages [Champaney, 1996, Blanzé et al., 2000].

Le cadre non-incrémental de la méthode Latin permet de mettre en place des techniques de séparation temps / espace adaptées aux méthodes de réduction de modèle [Ladevèze, 1985, Cognard and Ladevèze, 1993]. La linéarité de l'étape globale autorise à rechercher la solution sous la forme de modes à variables séparées. Cela permet de réduire la taille du problème par projection sur les modes et ainsi théoriquement réduire le temps de calcul et les besoins en mémoire. Des applications dans le cadre non-linéaire de la plasticité et décomposition de domaine peuvent être trouvées dans [Ladevèze et al., 2010, Passieux, 2009, Relun et al., 2013]. [Vitse, 2017] a proposé un couplage de la PGD avec la Latin pour des problèmes d'endommagement de structures en béton armé en incluant des paramètres dans la formulation PGD.

1.3 Méthode Latin multi-paramétrique

D'autres applications à des études paramétriques peuvent être trouvées dans [Boucard and Champaney, 2003, Odièvre, 2009, Roulet et al., 2011]. Lorsqu'une étude paramétrique implique

un paramètre présent dans l'espace non linéaire \mathcal{L} , il est possible de mettre en place un schéma multi-paramétrique. L'espace linéaire étant indépendant des paramètres, il est possible d'initialiser par une solution déjà déterminée, celle-ci étant dans \mathcal{A} . Concrètement, une première solution S_1 associée au premier jeu de paramètres est calculée en effectuant une initialisation classique par $S_0 = 0$. La solution suivante S_2 associée au deuxième jeu de paramètres est déterminée en utilisant la solution S_1 comme initialisation.

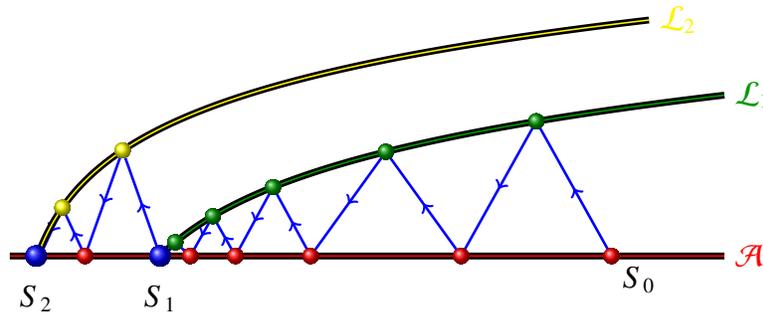


FIGURE 2.2 – Principe de la Latin multi-paramétrique

En initialisant par une solution dont le problème est proche de celui à résoudre, la convergence est grandement améliorée. Un illustration est présentée au chapitre 4.

2 Méthode Latin appliquée à la décomposition de domaine mono-échelle

2.1 Obtention du problème

Nous nous penchons désormais plus particulièrement sur la méthode Latin appliquée à la décomposition de domaine. Nous considérons uniquement pour ce chapitre des interfaces parfaites. Nous reprenons donc le problème 1.31 présenté en 3.1.4. Nous sommes sous les hypothèses d'élasticité linéaire en HPP, la connaissance du problème (champ de déplacement \mathbf{u} intérieur aux sous-domaines) peut alors se ramener à la connaissance des grandeurs d'interface (\mathbf{W}, \mathbf{F}) . Ainsi dans ce contexte de décomposition de domaine les solutions S_i s'écriront simplement par $S_i = [\mathbf{W}, \mathbf{F}]$. Nous choisissons pour ce chapitre de rester en statique afin de simplifier les notations. Le cas de la quasi-statique sera introduit au sein du chapitre 4.

$$\begin{aligned}
 \mathbf{K}\mathbf{u} &= \mathbf{f}_d + \mathbf{N}^T \mathbf{F} \\
 \mathbf{W} &= \mathbf{N}\mathbf{u} \\
 \mathbf{A}\mathbf{F} &= 0 \\
 \mathbf{B}\mathbf{W} &= 0
 \end{aligned} \tag{2.1}$$

Dans ce cadre la séparation des équations consiste à distinguer ce qui fait référence aux interfaces de ce qui est relatif aux sous-domaines. L'espace local regroupe donc les solutions vérifiant

les relations d'équilibre et de continuité des grandeurs d'interface. L'espace linéaire est quant à lui composé des solutions vérifiant l'équilibre des sous-domaines.

$$\begin{aligned} \mathcal{A} : (\mathbf{W}, \mathbf{F}) \text{ tel que } & \begin{cases} \mathbf{K}\mathbf{u} = \mathbf{f}_d + \mathbf{N}^T \mathbf{F} \\ \mathbf{W} = \mathbf{N}\mathbf{u} \end{cases} \\ \mathcal{L} : (\widehat{\mathbf{W}}, \widehat{\mathbf{F}}) \text{ tel que } & \begin{cases} \mathbf{A}\widehat{\mathbf{F}} = 0 \\ \mathbf{B}\widehat{\mathbf{W}} = 0 \end{cases} \end{aligned} \quad (2.2)$$

Les relations de direction de recherche portent elles aussi seulement sur les quantités d'interface. Elle définissent une relation linéaire entre les déplacements et les efforts d'interface à l'aide des opérateurs de direction de recherche \mathbf{k}^+ et \mathbf{k}^- .

$$\begin{aligned} k^+ : \widehat{\mathbf{F}} - \mathbf{F} - \mathbf{k}^+ (\widehat{\mathbf{W}} - \mathbf{W}) &= 0 \\ k^- : \mathbf{F} - \widehat{\mathbf{F}} + \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) &= 0 \end{aligned} \quad (2.3)$$

En mettant en place le schéma itératif, les deux étapes successives de la méthode Latin deviennent :

Etape locale : connaissant $S_i = (\mathbf{F}_i, \mathbf{W}_i) \in \mathcal{A}$, trouver $\widehat{S}_i = (\widehat{\mathbf{F}}_i, \widehat{\mathbf{W}}_i) \in \mathcal{L}$ tel que :

$$\begin{aligned} \mathbf{A}\widehat{\mathbf{F}}_i &= 0 \\ \mathbf{B}\widehat{\mathbf{W}}_i &= 0 \\ \widehat{\mathbf{F}}_i - \mathbf{F}_i - \mathbf{k}^+ (\widehat{\mathbf{W}}_i - \mathbf{W}_i) &= 0 \end{aligned} \quad (2.4)$$

Etape linéaire : connaissant $\widehat{S}_i = (\widehat{\mathbf{F}}_i, \widehat{\mathbf{W}}_i) \in \mathcal{L}$, trouver $S_{i+1} = (\mathbf{F}_{i+1}, \mathbf{W}_{i+1}) \in \mathcal{A}$ tel que :

$$\begin{aligned} \mathbf{K}\mathbf{u}_{i+1} &= \mathbf{f}_d + \mathbf{N}^T \mathbf{F}_{i+1} \\ \mathbf{W}_{i+1} &= \mathbf{N}\mathbf{u}_{i+1} \\ \mathbf{F}_{i+1} - \widehat{\mathbf{F}}_i + \mathbf{k}^- (\mathbf{W}_{i+1} - \widehat{\mathbf{W}}_i) &= 0 \end{aligned} \quad (2.5)$$

Remarque 2.3

Le fait d'introduire une direction de recherche qui ne privilégie ni les champs de déplacement ni les champs d'effort d'interface impose de gérer des conditions mixtes sur le bord des sous-domaines. Cela conduit à considérer la méthode Latin appliquée à la décomposition de domaine comme une approche mixte.

2.2 Convergence

Une condition de convergence est de respecter $\mathbf{k}^- = \mathbf{k}^+$. Sans étape de relaxation, [Ladevèze, 1999] montre que la suite $\left(\frac{S_{i+1}+S_i}{2}\right)_i$ converge vers la solution du problème. Avec l'étape de relaxation la suite $(S_i)_i$ converge vers la solution du problème quelle que soit la valeur du coefficient de relaxation comprise strictement entre 0 et 1.

Un indicateur de convergence classique est choisi et représente l'écart entre la solution déterminée lors de l'étape locale et la solution déterminée lors de l'étape linéaire.

$$\eta = \frac{\|S_{i+1} - \widehat{S}_i\|^2}{\|S_{i+1}\|^2 + \|\widehat{S}_i\|^2} = \frac{(\mathbf{F} - \widehat{\mathbf{F}})^T \mathbf{k}^{-1} (\mathbf{F} - \widehat{\mathbf{F}}) + (\mathbf{W} - \widehat{\mathbf{W}})^T \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}})}{\mathbf{F}^T \mathbf{k}^{-1} \mathbf{F} + \widehat{\mathbf{F}}^T \mathbf{k}^{-1} \widehat{\mathbf{F}} + \mathbf{W}^T \mathbf{k}^- \mathbf{W} + \widehat{\mathbf{W}}^T \mathbf{k}^- \widehat{\mathbf{W}}} \quad (2.6)$$

Cet indicateur d'erreur fait intervenir l'opérateur de direction de recherche. Il est donc nécessaire de faire attention lors des études d'influence de la direction de recherche sur la convergence. Cet indicateur peut devenir biaisé et d'autres grandeurs doivent être considérées. La quantité de saut de déplacement à l'étape linéaire ou encore le déséquilibre d'effort peuvent être des indicateurs pertinents qui ne dépendent pas de la direction de recherche. Qui plus est, l'indicateur de convergence classique ne donne qu'une information globale des solutions. Des perturbations très localisées pourraient ne pas être détectées.

2.3 Interprétation des directions de recherches

La convergence de la méthode est fortement liée au choix de l'opérateur de direction de recherche. Nous considérons une sous-structure Ω_E et \mathbf{k}_E l'opérateur de direction de recherche associé aux interfaces de Ω_E . Le choix optimal de direction de recherche est l'opérateur qui représente au mieux le comportement du complémentaire $\overline{\Omega}_E = \Omega \setminus \Omega_E$ de Ω_E dans Ω . Cet opérateur n'est rien d'autre que le complément de Schur du complémentaire de la sous-structure condensé sur l'interface de Ω_E . Cependant il n'est pas possible d'avoir accès à un tel opérateur. Plusieurs raisons peuvent être avancées :

- le complémentaire de la sous-structure est de grande taille et par conséquent l'inversion de l'opérateur $\mathbf{K}_{\overline{\Omega}_E}$ nécessaire au calcul du complément de Schur a un coup énorme.
- dans un contexte parallèle, les sous-domaines n'ont généralement accès qu'à des informations les concernant. Le calcul d'un tel opérateur nécessiterait des communications globales importantes entre les sous-domaines.

Une approximation doit donc être faite. De récents travaux [Negrello et al., 2017] dans le cadre d'approches mixtes proposent une décomposition de cet opérateur en un terme relatif aux sous-domaines voisins et un terme relatif aux sous-domaines plus lointains. Cette séparation d'échelle a pour objectif de représenter au mieux l'action du complémentaire de la sous-structure tout en imposant un caractère creux pour l'opérateur de direction de recherche.

Classiquement les différents travaux sur la méthode Latin proposent un choix simple d'opérateur de direction de recherche. Ecrite sous forme faible la direction de recherche devient :

$$\forall \Gamma_{EE'} \in \mathcal{G}, \int_{\Gamma_{EE'}} (F_{EE'} - \widehat{F}_{EE'}) W^* dS \pm \int_{\Gamma_{EE'}} k_{EE'} (W_{EE'} - \widehat{W}_{EE'}) W^* dS = 0 \quad (2.7)$$

et ainsi l'opérateur est choisi scalaire et valant $k_{EE'} = E/L$ avec E le module de Young et L une longueur caractéristique de la structure. Ce choix représente une rigidité de la structure et a pour objectif de représenter simplement le comportement global de la structure.

Remarque 2.4

- Un seul paramètre est utilisé dans le cas classique de la méthode Latin pour paramétrer l'ensemble des interfaces.
- Ce choix classique de direction de recherche rarement remis en cause pose de grandes difficultés pour des structures élancées et doit absolument être adapté pour assurer la convergence de la méthode. Des travaux très récents de [Saavedra et al., 2017] ont proposé une étude de cette direction de recherche pour des phénomènes de flambage de structures élancées.

L'opérateur de direction de recherche doit être défini comme un opérateur de $H^{1/2}(\Gamma_{EE'})$ sur son dual pour assurer la bonne définition des efforts d'interface. Cependant, il n'est pas nécessairement local tel qu'il est défini pour la méthode Latin standard. Une proposition de direction de recherche non-locale a été formulée par [Desmeure, 2016]. Cela consiste à écrire $k_{EE'}$ sous la forme :

$$k_{EE'} = k_{EE'}^* J_{EE'} \quad (2.8)$$

avec $J_{EE'}$ un isomorphisme de Riesz de $H^{1/2}(\Gamma_{EE'})$ dans son dual et $k_{EE'}^*$ choisi comme étant scalaire. Cette formulation, permettant une meilleure représentation des efforts, offre une convergence plus rapide de la méthode. Cependant il faut noter une implémentation beaucoup plus complexe qui ne peut rentrer dans un cadre non-intrusif.

2.4 Etape linéaire intrusive

Les difficultés de l'implémentation de la méthode Latin proviennent principalement de l'étape linéaire. Le système 2.5 conduit au problème en déplacement suivant :

$$(\mathbf{K} + \mathbf{N}^T \mathbf{k}^- \mathbf{N}) \mathbf{u} = \mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) \quad (2.9)$$

Les grandeurs d'interface sont ensuite déterminées avec :

$$\begin{aligned} \mathbf{W} &= \mathbf{N} \mathbf{u} \\ \mathbf{F} &= \widehat{\mathbf{F}} - \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) \end{aligned} \quad (2.10)$$

L'équation 2.9 peut être interprétée comme la discrétisation du problème d'équilibre sur les sous-domaines sous des conditions de Robin généralisées. Le terme $\mathbf{N}^T \mathbf{k}^- \mathbf{N}$ est une impédance d'interface qu'il n'est pas aisé de calculer au sein des codes industriels. Le choix classique de direction de recherche conduit après discrétisation à un opérateur $\mathbf{k}_{EE'}$ de la forme :

$$\mathbf{k}_{EE'} = \int_{\Gamma_{EE'}} \phi^T k_{EE'} \phi dS = \int_{\Gamma_{EE'}} k_{EE'} \phi^T \phi dS \quad (2.11)$$

Cette impédance d'interface est liée à l'intégration du produit matriciel des fonctions de forme. Cela diffère d'un opérateur de rigidité pour lequel interviennent les dérivées des fonctions de forme éléments finis. Un tel calcul de rigidité n'est donc pas implémenté au sein des codes industriels. Cet opérateur devrait plutôt être assimilé à un opérateur de masse calculable quant à lui au sein de ces codes industriels. Cependant l'assemblage de ces opérateurs \mathbf{K} et $\mathbf{N}^T \mathbf{k}^{-1} \mathbf{N}$ n'y est pas prévu. Il est donc nécessaire de proposer une formulation de la direction de recherche qui permette une telle opération.

3 Vision non-intrusive par direction de recherche non-locale

3.1 Direction de recherche non-locale

3.1.1 Principe de la direction de recherche non intrusive

Nous proposons une nouvelle implémentation des conditions de Robin d'interface par l'ajout d'une couche d'éléments au niveau des interfaces des sous-domaines. La Figure 2.3 présente une structure composée de trois sous-domaines et deux interfaces. Nous nous focalisons sur le sous-domaine 1. Au niveau de ses interfaces avec les sous-domaines voisins on ajoute une couche d'éléments que nous appellerons *semelle* par la suite. Plusieurs couches d'éléments pourraient être ajoutées. Dans la pratique une seule suffit pour gérer les conditions de Robin.

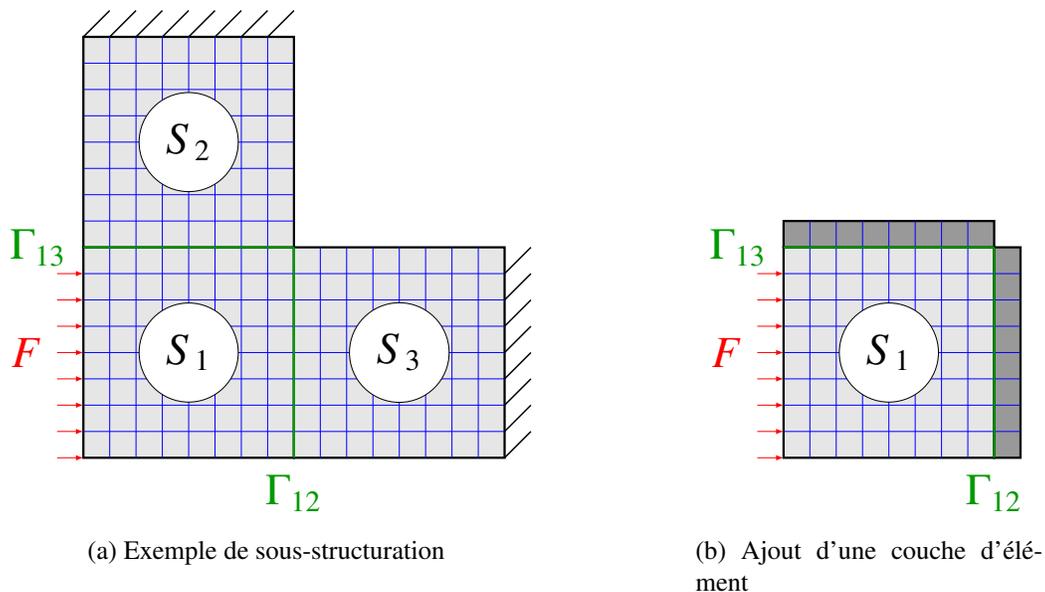


FIGURE 2.3 – Exemple de couche d'élément aux interfaces

L'idée est d'utiliser cette couche d'éléments tel un "ressort" pour transformer une impédance d'interface complexe à déterminer en une rigidité éléments finis calculable facilement dans tout code industriel. Cette direction de recherche peut être qualifiée de non-locale dans le sens où les noeuds d'une interface deviennent couplés à travers la semelle associée.

Une fois la couche d'éléments définie, l'opérateur de direction de recherche est paramétré par les deux grandeurs que sont les paramètres matériaux E et ν : module de Young et coefficient de Poisson de la semelle. Dans les faits, le coefficient de Poisson sera identique à celui de la structure et la paramètre réellement considéré sera le module de Young.

Remarque 2.5

- Des éléments type ressort noeud à noeud auraient pu être utilisés pour prendre en compte ces conditions de Robin. Ce type de rigidité est généralement implémenté dans les codes industriels.
- Un projet de première année de master a montré que l'utilisation d'une couche d'éléments cohésifs sollicités dans leur domaine élastique linéaire permet aussi de gérer ces conditions de Robin. Cependant le code cible, *code_aster*, force l'utilisation d'un solveur non-linéaire (qui ne conserve pas les factorisations) quand ces éléments sont utilisés, ce qui détériore fortement les performances.

3.1.2 Direction de recherche non-intrusive via un problème de Dirichlet

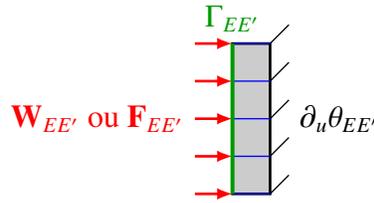
Cet ajout d'éléments au niveau des interfaces permet de définir la relation de direction de recherche comme étant un problème de Dirichlet sur les semelles. En effet, en notant $\theta_{EE'}$ la semelle ajoutée au sous-domaine Ω_E relative à l'interface $\Gamma_{EE'}$, $\mathbb{K}_{EE'}$ le tenseur de Hooke associé à son comportement mécanique, l'opérateur $k_{EE'}$ est défini de façon continue par la relation suivante :

$$F_{EE'} = k_{EE'}(W_{EE'}) \Leftrightarrow \begin{cases} u \in \mathcal{H}^1(\Omega)(\theta_{EE'}), u = W_{EE'} \text{ sur } \Gamma_{EE'}, u = 0 \text{ sur } \partial_u \theta_{EE'} \\ \forall v \in \mathcal{H}^1(\Omega)(\theta_{EE'}), v = 0 \text{ sur } \Gamma_{EE'} \cap \partial_u \theta_{EE'} \\ \int_{\theta_{EE'}} \epsilon(u) : \mathbb{K}_{EE'} : \epsilon(v) d\Omega = \int_{\Gamma_{EE'}} F_{EE'} \cdot v dS \end{cases} \quad (2.12)$$

avec $\partial_u \theta_{EE'}$ le bord qui n'est pas en contact avec l'interface $\Gamma_{EE'}$.

D'un point de vue discret cet opérateur de direction de recherche $\mathbf{k}_{EE'}$ renvoie les réactions nodales $\mathbf{F}_{EE'}$ à l'interface $\Gamma_{EE'}$ pour un chargement de Dirichlet imposé $\mathbf{W}_{EE'}$. Inversement l'opérateur $\mathbf{k}_{EE'}^{-1}$ renvoie le déplacement $\mathbf{W}_{EE'}$ sur l'interface $\Gamma_{EE'}$ pour un chargement de Neumann imposé $\mathbf{F}_{EE'}$.

Ainsi, si on note $\mathbf{K}_{EE'}$ l'opérateur de rigidité de la semelle encastrée sur son bord $\partial_u \theta_{EE'}$, alors l'opérateur de direction de recherche $\mathbf{k}_{EE'}$ n'est rien d'autre que le complément de Schur de $\mathbf{K}_{EE'}$ condensé sur l'interface $\Gamma_{EE'}$.

FIGURE 2.4 – Problème de Dirichlet ou Neumann sur $\theta_{EE'}$

⚠ Remarque 2.6

- On pourrait être tenté de dresser un parallèle avec la méthode restricted additive Schwarz [Cai and Sarkis, 1999]. Cependant dans le cas non intrusif proposé, cet ajout d'éléments n'est effectué que pour traduire des conditions mixtes sur les interfaces et relier de façon homogène les efforts et les déplacements. En aucun cas des conditions de Dirichlet différentes de l'encastrement sont appliquées sur le bord $\partial_u \theta_{EE'}$ contrairement aux méthodes de Schwarz.
- A l'aide de ces semelles, la direction de recherche assure une définition correcte des efforts. En effet ces efforts sont nécessairement homogènes à des réactions nodales extraites via le code industriel. Ces efforts peuvent ensuite être directement appliqués comme conditions limites.

3.1.3 Etape linéaire non-intrusive

A l'aide de ce nouvel opérateur de direction de recherche, \mathbf{k}^- est transformé en un véritable opérateur de rigidité éléments finis. Cela permet de traiter le premier membre du système 2.9 de l'étape linéaire comme un assemblage de rigidité. Ainsi $\mathbf{K} + \mathbf{N}^T \mathbf{k}^- \mathbf{N}$ correspond à la rigidité du sous-domaine auquel est rajouté l'ensemble de ses semelles. Le second membre $\mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}})$ est un saut d'effort nodal à appliquer aux noeuds d'interface. Le problème à résoudre à l'étape linéaire est illustré à la figure 2.5.

⚠ Remarque 2.7

L'objectif est d'ajouter le moins d'éléments possibles aux interfaces pour ne pas augmenter la taille du problème à résoudre. C'est pourquoi il est choisi de n'ajouter qu'une couche d'éléments aux interfaces. Ce choix là nous permet même de ne pas du tout augmenter la taille du problème pour des éléments linéaires. En effet, les noeuds qui ne sont pas en commun avec l'interface sont encastres, et une prise en compte de ces conditions par élimination supprime les degrés de liberté associés.

$$(\mathbf{K} + \mathbf{N}^T \mathbf{k}^- \mathbf{N}) \mathbf{u} = \mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) \quad (2.13)$$

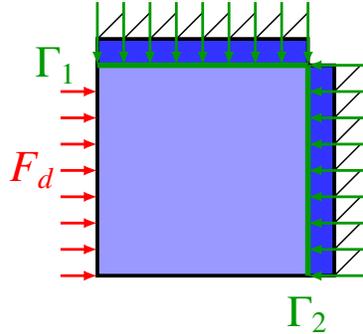


FIGURE 2.5 – Illustration de l'étape linéaire

Le champ de déplacement d'interface \mathbf{W} est directement obtenu à partir du champ de déplacement \mathbf{u} comme étant une extraction sur les noeuds d'interface. Le champ d'effort est quant à lui déterminé par la relation de direction de recherche :

$$\mathbf{F} = \widehat{\mathbf{F}} - \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) \quad (2.14)$$

⚠ Remarque 2.8

Comme expliqué à la section 3.1.2, l'application de l'opérateur de direction de recherche à un champ de déplacement se traduit par la résolution d'un problème de Dirichlet sur les semelles. Cependant il est possible de post-traiter directement le résultat $\mathbf{k}^- \mathbf{W}$ du calcul sur les sous-domaines et les semelles. En effet lorsqu'il est résolu ce problème $\mathbf{k}^- \mathbf{W}$ n'est rien d'autre que la réaction nodale calculée sur les semelles au niveau des interfaces. Il n'est donc pas nécessaire de venir résoudre des problèmes intermédiaires sur les semelles pour utiliser la relation de direction de recherche.

3.1.4 Etape locale non-locale

Initialement le problème d'intrusivité ne provenait que de la direction de recherche de l'étape linéaire. Celle-ci a été choisie comme étant non-locale avec un couplage des noeuds d'interface. Il est aussi possible de faire de même lors de l'étape locale sur les interfaces. Cela permet de garantir $\mathbf{k}^+ = \mathbf{k}^-$ et vérifier les conditions de convergence de la méthode Latin.

L'étape locale consiste à déterminer $(\widehat{\mathbf{W}}, \widehat{\mathbf{F}})$ connaissant (\mathbf{W}, \mathbf{F}) tel que :

$$\begin{aligned} \mathbf{A}\widehat{\mathbf{F}} &= 0 \\ \mathbf{B}\widehat{\mathbf{W}} &= 0 \\ \widehat{\mathbf{F}} - \mathbf{F} - \mathbf{k}^+ (\widehat{\mathbf{W}} - \mathbf{W}) &= 0 \end{aligned} \quad (2.15)$$

Les conditions d'équilibre et de continuité des champs d'interface imposent des formes particulières de solutions :

$$\begin{aligned} \mathbf{A}\widehat{\mathbf{F}} &= 0 & \widehat{\mathbf{F}} &= \mathbf{B}^T \mathbf{F}^* \\ \mathbf{B}\widehat{\mathbf{W}} &= 0 & \widehat{\mathbf{W}} &= \mathbf{A}^T \mathbf{W}^* \end{aligned} \quad (2.16)$$

avec \mathbf{F}^* et \mathbf{W}^* définis sur les interfaces et non plus sur les bords des sous-domaines.

Ainsi en injectant ces formulations dans la relation de direction de recherche et en appliquant soit \mathbf{A} , soit \mathbf{B} à cette relation, il en est déduit l'expression des champs de déplacement et d'effort :

$$\begin{aligned} \widehat{\mathbf{F}} &= \mathbf{B}^T (\mathbf{B}\mathbf{k}^{+-1}\mathbf{B}^T)^{-1} [\mathbf{B}\mathbf{k}^{+-1}\mathbf{F} - \mathbf{B}\mathbf{W}] \\ \widehat{\mathbf{W}} &= \mathbf{A}^T (\mathbf{A}\mathbf{k}^+\mathbf{A}^T)^{-1} [\mathbf{A}\mathbf{k}^+\mathbf{W} - \mathbf{A}\mathbf{F}] \end{aligned} \quad (2.17)$$

En considérant ces relations à l'interface $\Gamma_{EE'}$, les champs solutions deviennent :

$$\begin{aligned} \widehat{\mathbf{F}}_{EE'} &= -\widehat{\mathbf{F}}_{E'E} = -(\mathbf{k}_{EE'}^{+-1} + \mathbf{k}_{E'E}^{+-1})^{-1} [\mathbf{W}_{EE'} - \mathbf{W}_{E'E} + \mathbf{k}_{E'E}^{+-1}\widehat{\mathbf{F}}_{E'E} - \mathbf{k}_{EE'}^{+-1}\widehat{\mathbf{F}}_{EE'}] \\ \widehat{\mathbf{W}}_{EE'} &= \widehat{\mathbf{W}}_{E'E} = (\mathbf{k}_{EE'}^+ + \mathbf{k}_{E'E}^+)^{-1} [\mathbf{k}_{EE'}^+\mathbf{W}_{EE'} + \mathbf{k}_{E'E}^+\mathbf{W}_{E'E} - \mathbf{F}_{EE'} - \mathbf{F}_{E'E}] \end{aligned} \quad (2.18)$$

L'opérateur $(\mathbf{k}_{EE'}^+ + \mathbf{k}_{E'E}^+)$ implique que le champ de déplacement $\widehat{\mathbf{W}}_{EE'}$ est obtenu à l'aide d'une résolution d'un problème de Neumann sur les deux semelles assemblées. Le chargement appliqué à l'interface entre les deux semelles ne dépend que des grandeurs connues de l'étape linéaire. (Figure 2.6).

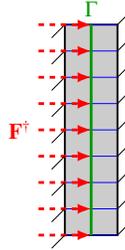


FIGURE 2.6 – Problème sur les doubles semelles : $\mathbf{F}^\dagger = \mathbf{A}\mathbf{k}^+\mathbf{W} - \mathbf{A}\mathbf{F}$

Une fois le champ de déplacement déterminé via le problème sur les doubles semelles, le champ d'effort est déterminé à l'aide de la relation de direction de recherche.

⚠ Remarque 2.9

De la même façon que lors de l'étape linéaire, le terme $\mathbf{k}^+\widehat{\mathbf{W}}$ est post-traité directement du calcul sur la double couche d'éléments comme étant la réaction nodale sur l'une des semelles à l'interface.

Les noeuds qui ne sont pas en commun avec l'interface sont encastres ce qui conduit après élimination des degrés de liberté à un problème de taille $n_\Gamma \times n_\Gamma$ avec n_Γ le nombre de degrés de liberté associés aux noeuds d'interface.

3.1.5 Prise en compte des conditions limites

Approche classique : les conditions limites sont prises en compte en introduisant de nouvelles interfaces entre les sous-structures et des "extérieurs". Cela permet de transformer les conditions imposées sur les sous-domaines en conditions sur des grandeurs d'interface des "extérieurs" lors de l'étape linéaire.

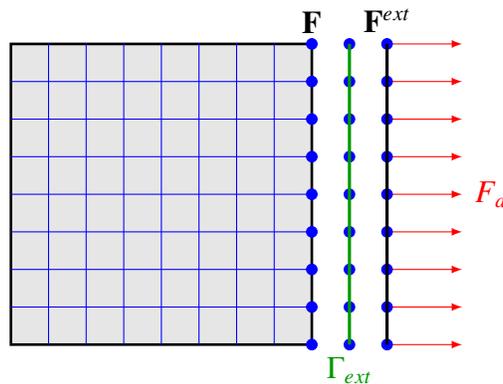


FIGURE 2.7 – Prise en compte classique des conditions limites

L'approche choisie : Pour des questions d'implémentation nous choisissons d'imposer directement les conditions limites sur les sous-domaines lors de l'étape linéaire. Aucune autre interface n'est alors introduite que celles entre les sous-domaines. Les conditions limites sont alors vérifiées de manière exacte à chaque itération.

3.2 Validation sur un cas simple 1D

3.2.1 Comparaison avec une implémentation classique

L'objectif de cet exemple est de valider l'implémentation non-intrusive en la comparant à une implémentation classique des conditions de Robin aux interfaces. Pour cela nous utilisons un cas simple de poutre en traction de module de Young E , de longueur L et de section S (Figure 2.8). Une décomposition en cinq sous-domaines est choisie. L'implémentation classique est faite sous *Matlab*. L'implémentation non-intrusive est quant à elle effectuée sous *code_aster*. Un modèle 2D dans *code_aster* avec un coefficient de Poisson nul est réalisé pour être équivalent à un cas 1D.

L'avantage de ce premier cas simple est de réduire la direction de recherche non-locale à un unique paramètre : le module de Young des semelles. Nous rappelons que dans la méthode Latin standard, cette direction de recherche est $\mathbf{k}^+ = \mathbf{k}^- = \frac{ES}{L}$. Ainsi pour avoir une direction de recherche

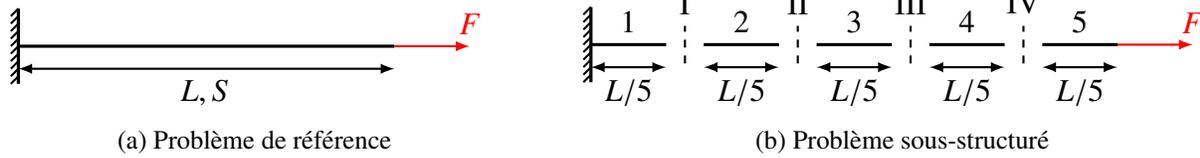


FIGURE 2.8 – Poutre en traction

équivalente, il convient simplement d'adapter le module de Young des semelles pour avoir une rigidité égale à $\frac{ES}{L}$. Soit un module de Young des semelles :

$$E_S = E \frac{L_S}{L} \quad (2.19)$$

avec L_S la longueur caractéristique des semelles qui n'est autre que la taille de maille. Les caractéristiques du calcul sont données à la table 2.1.

Paramètres	Valeurs
E	200 GPa
L	100 mm
S	10 mm
$h=L_S$	1 mm
F	10 MPa

TABLE 2.1 – Paramètres

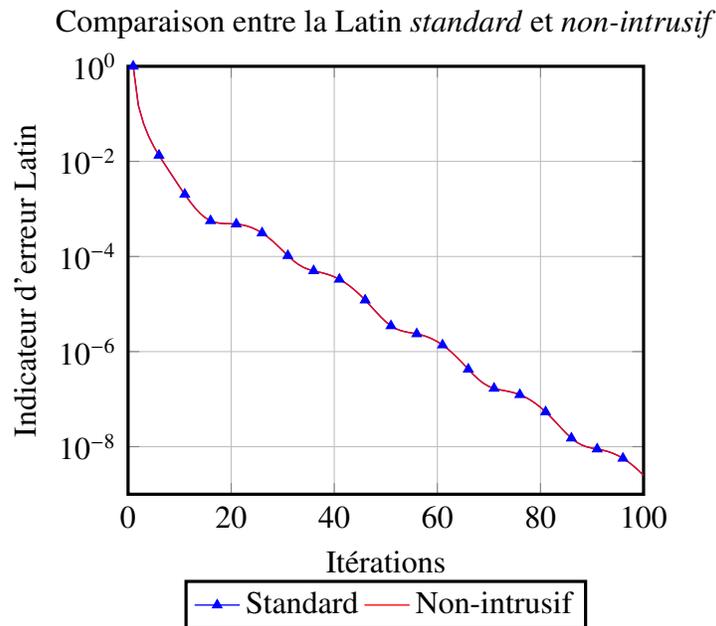
La figure 2.9 illustre la correspondance parfaite entre l'implémentation *standard* et l'implémentation *non-intrusive*. Il est en effet normal d'obtenir des résultats identiques car avec ce modèle 1D la direction de recherche non-intrusive est appliquée pour chaque interface à un noeud unique. Le caractère non-local n'existe donc pas et le choix du module de Young permet d'obtenir une direction de recherche identique au cas standard.

Nous illustrons l'évolution des déformées des sous-domaines au cours des itérations à la figure 2.10. L'influence des conditions limites se propage petit à petit d'un sous-domaine à l'autre.

3.2.2 Influence des directions de recherche

Nous proposons maintenant une rapide étude de l'influence des directions de recherche sur la convergence de la méthode. Nous considérons les directions de recherche telles que $\mathbf{k}^- = \mathbf{k}^+ = \alpha \frac{ES}{L}$ avec $\alpha > 0$. Nous choisissons de faire varier α entre 0.01 et 100. Nous présentons les résultats de cette étude pour différentes sous-structurations. La figure 2.11 montre qu'il existe une direction de recherche optimale qui ne dépend pas de la sous-structuration. Cela confirme le résultat classique de la méthode Latin : la direction de recherche optimale ne dépend que de la structure globale.

Il est précisé à la section 2.3 que l'indicateur d'erreur classique de la méthode Latin fait intervenir la direction de recherche. Il peut donc sembler illégitime de tirer des conclusions de ces


 FIGURE 2.9 – Comparaison entre Latin *standard* et *non-intrusif*

résultats. Nous choisissons alors deux autres indicateurs qui ne font pas intervenir la direction de recherche et qui se focalisent respectivement sur le saut de déplacement aux interfaces à la fin de l'étape linéaire et sur le déséquilibre des efforts aux interfaces. Le fait de distinguer les déplacements des efforts nous autorise à ne plus faire intervenir la direction de recherche :

— Saut de déplacement aux interfaces :

$$\eta_{Saut} = \frac{(\mathbf{B}\mathbf{W})^T \mathbf{B}\mathbf{W}}{(\mathbf{A}\mathbf{W})^T \mathbf{A}\mathbf{W}} \quad (2.20)$$

— Déséquilibre aux interfaces :

$$\eta_{deseq} = \frac{(\mathbf{A}\mathbf{F})^T \mathbf{A}\mathbf{F}}{(\mathbf{B}\mathbf{F})^T \mathbf{B}\mathbf{F}} \quad (2.21)$$

Sur la figure 2.12 la direction de recherche optimale peut aussi être observée et confirme donc le résultat précédent. Cependant il convient de rester prudent quant aux résultats fournis par l'indicateur classique. Cet exemple simple peut ne pas être généralisable (exemple 4.3).

Remarque 2.10

Jusqu'ici la structure globale était considérée comme homogène. La pondération du module de Young ne faisait donc intervenir que le module de Young de la structure globale. Dans le cas où un sous-domaine a des caractéristiques matériau différentes des autres sous-domaines, il est

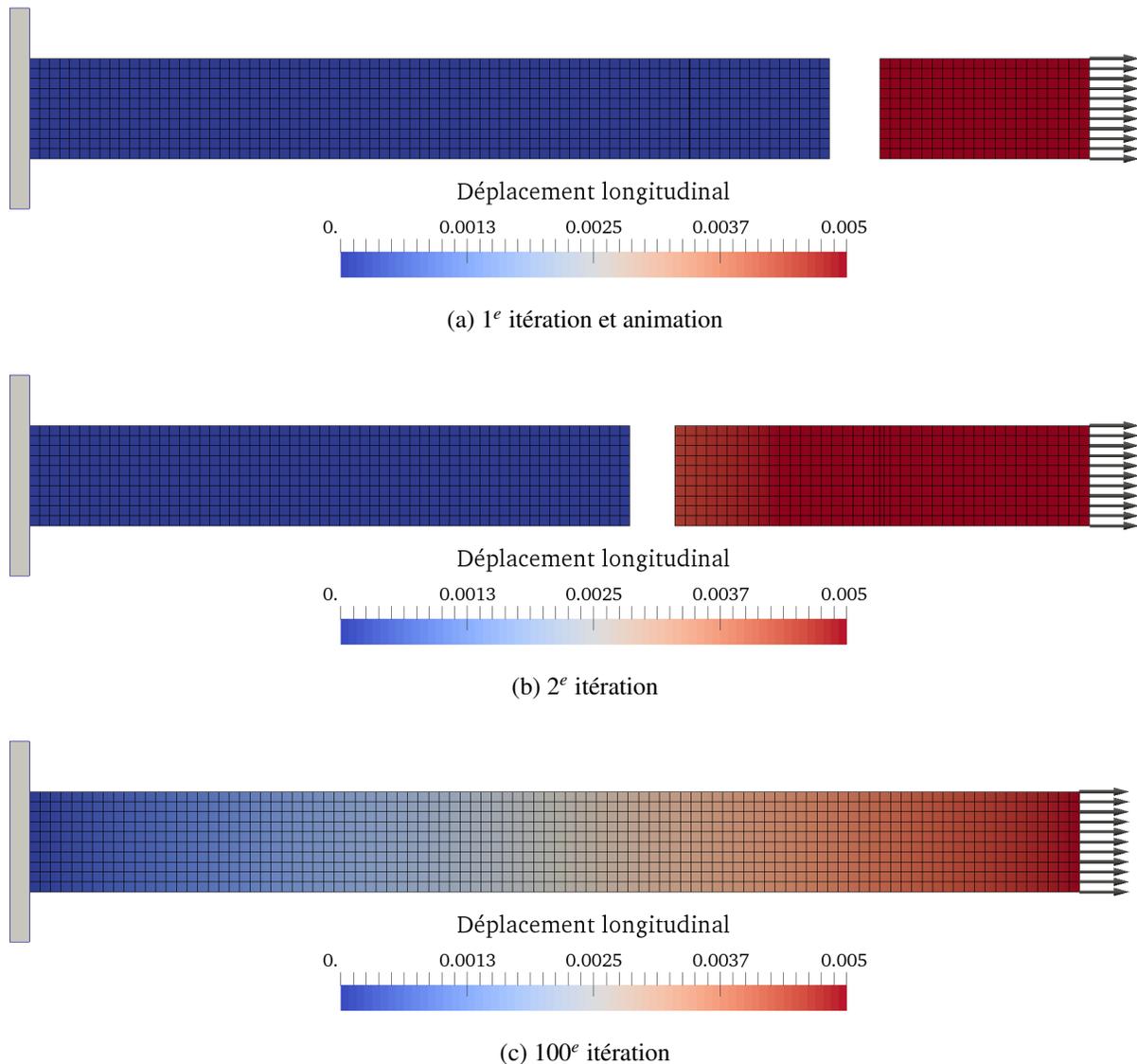


FIGURE 2.10 – Évolution de la déformée au cours des itérations

nécessaire de considérer un module "homogénéisé" de la structure afin d'éviter un trop fort saut de direction de recherche entre deux sous-structures.

3.3 Présentation d'une simulation 3D

Cet exemple a pour but de comparer dans un cas 3D une solution obtenue par la méthode Latin à un calcul de référence direct obtenu par *code_aster*. Une jonction de tuyau (Figure 2.13) est sous-structurée automatiquement en 30 sous-domaines par l'intermédiaire de *Scotch* [Pellegrini and Roman, 1996] inclus dans *code_aster*. Cette sous-structuration automatique permet d'étudier le comportement de la méthode Latin pour des cas d'interfaces non-régulières (Figure 2.14b). Ce

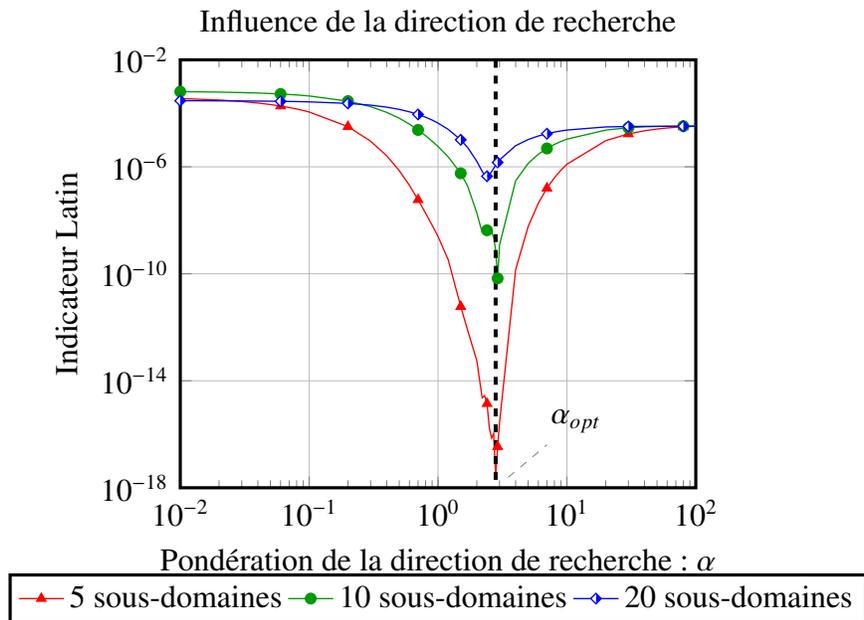


FIGURE 2.11 – Influence de la direction de recherche

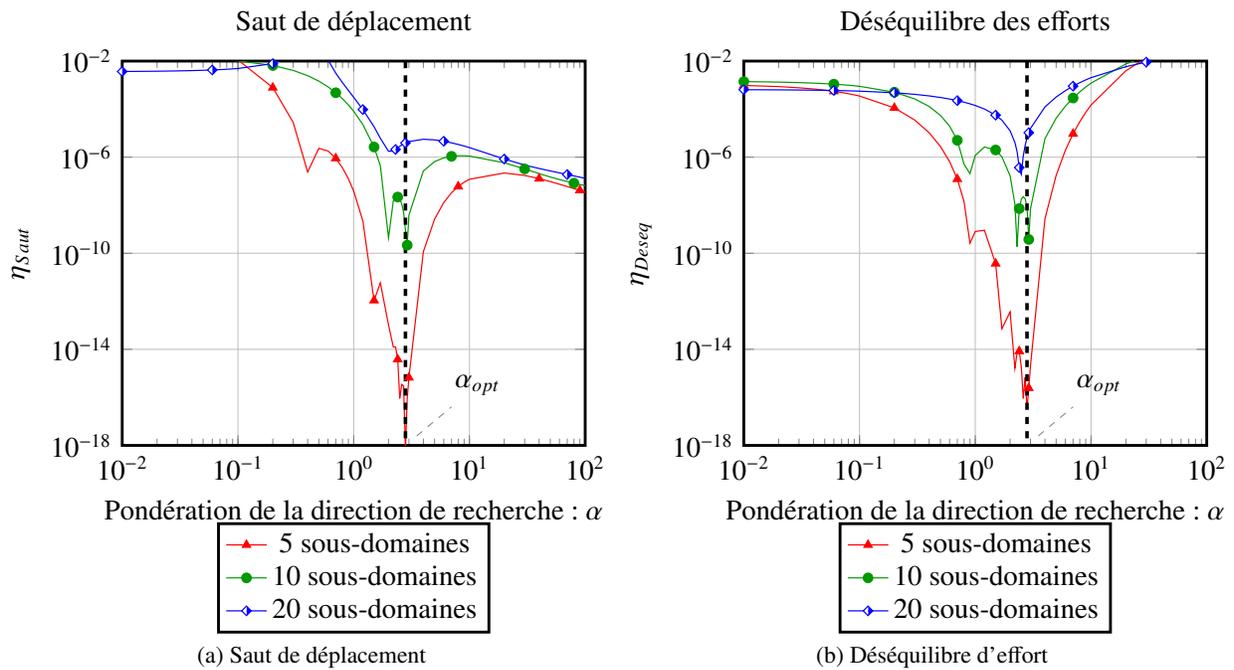


FIGURE 2.12 – Saut et déséquilibre aux interfaces

type d'interface n'a jusqu'ici pas été abordé par la littérature dans le cadre de la méthode Latin.

Les déplacements aux extrémités sont bloqués. Une pression interne de 1MPa est appliquée. Le module de Young est choisi à 200GPa et le coefficient de Poisson vaut 0.3. Le problème global

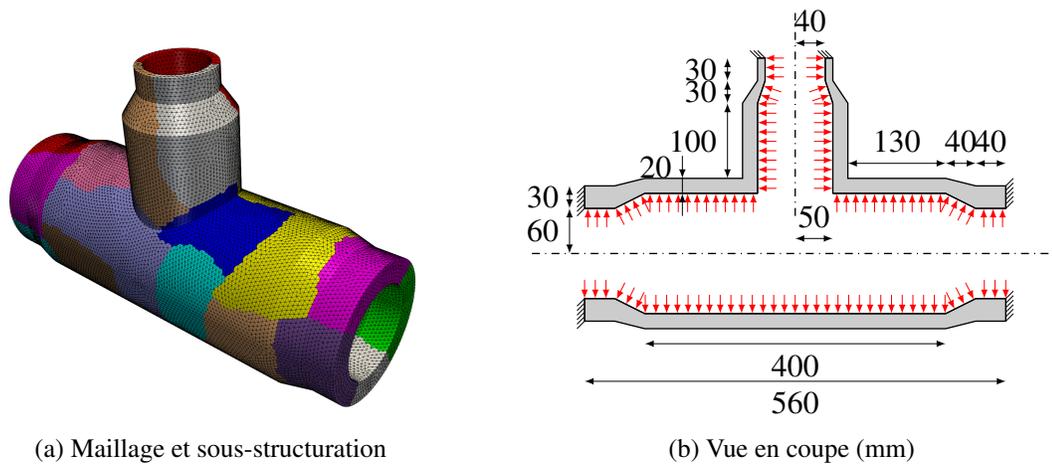


FIGURE 2.13 – Cas test 3D : jonction de tuyau

comporte 186 564 degrés de liberté. Le critère de convergence est choisi à 10^{-7} .

Paramètres	Valeurs
E	200 GPa
ν	0.3
pression	1 MPa
η	10^{-7}
ddl	186564

TABLE 2.2 – Paramètres du calcul de jonction 3D

Le critère de convergence est atteint au bout de 351 itérations. La convergence obtenue (Figure 2.15) est une convergence classique pour la méthode Latin. Nous représentons à la figure 2.16 la norme du déplacement sur la structure pour la solution obtenue par un calcul de référence direct et la solution obtenue par la méthode Latin. A titre d'exemple nous présentons les déformés obtenues à la première et cinquantième itérations à la figure 2.17. La pression interne fait se déformer l'ensemble des sous-domaines dès la première itération (Figure 2.17a). Les grandeurs d'interface étant initialisées à zéro les sous-domaines sont donc quasiment encastres aux interfaces. Cela explique l'aspect boursoufflé de la structure. Au cours des itérations les champs d'interface deviennent non nuls et les informations se transmettent de sous-domaines en sous-domaines.

Aux figures 2.18 est illustré l'écart entre ces deux solutions pour certaines itérations du calcul. Nous y représentons les zones pour lesquelles l'écart par rapport à la référence est au-dessus de 15%. Nous observons très nettement un rétrécissement de cette zone au cours des itérations. Cet écart qui semble tout de même persister au cours des itérations peut s'expliquer pour plusieurs raisons :

- la sous-structuration automatique produit des interfaces non-régulières (Figure 2.14b) ce qui est connu pour détériorer le conditionnement du problème [Klawonn et al., 2008].

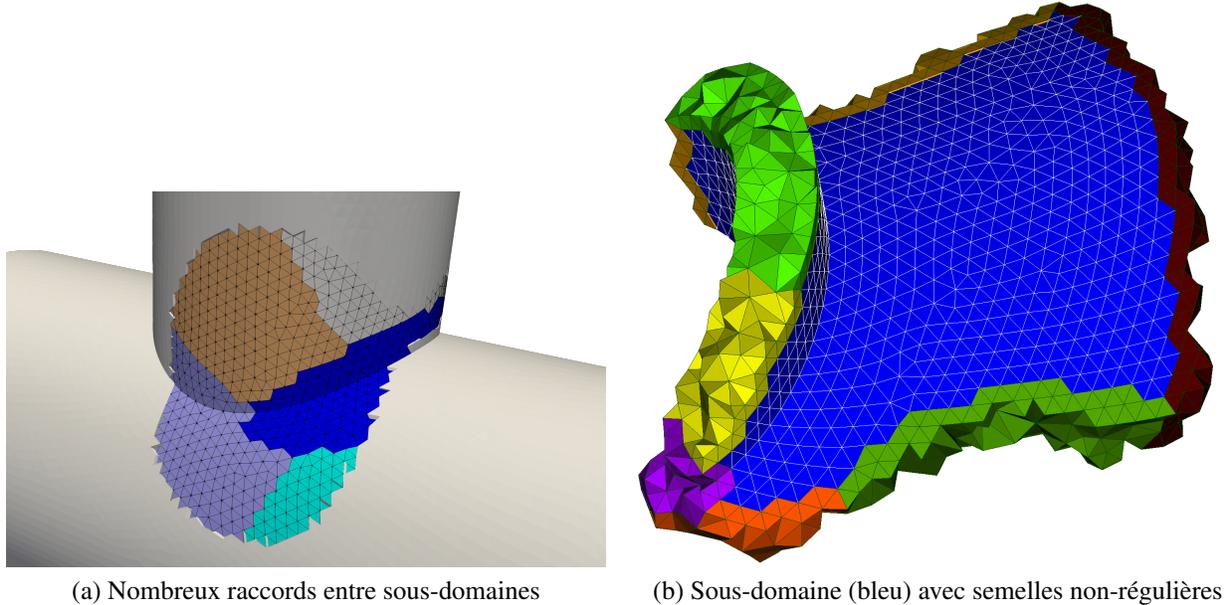


FIGURE 2.14 – Interfaces non-régulières et raccords multiples

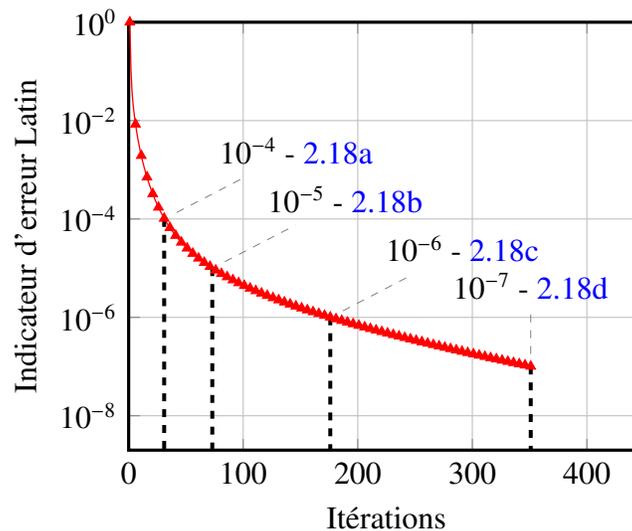


FIGURE 2.15 – Indicateur d'erreur Latin

— les raccords multiples entre sous-domaines ralentissent inévitablement la convergence de la méthode (Figure 2.14a),

Ainsi la performance se dégrade en présence d'interfaces irrégulières dues à une sous-structuration automatique. Nous éviterons donc par la suite d'avoir recours à une sous-structuration automatique pour laquelle l'utilisation de solveurs adaptés [Spillane and Rixen, 2013, Gosselet et al., 2015] est laissée en perspective de cette thèse.

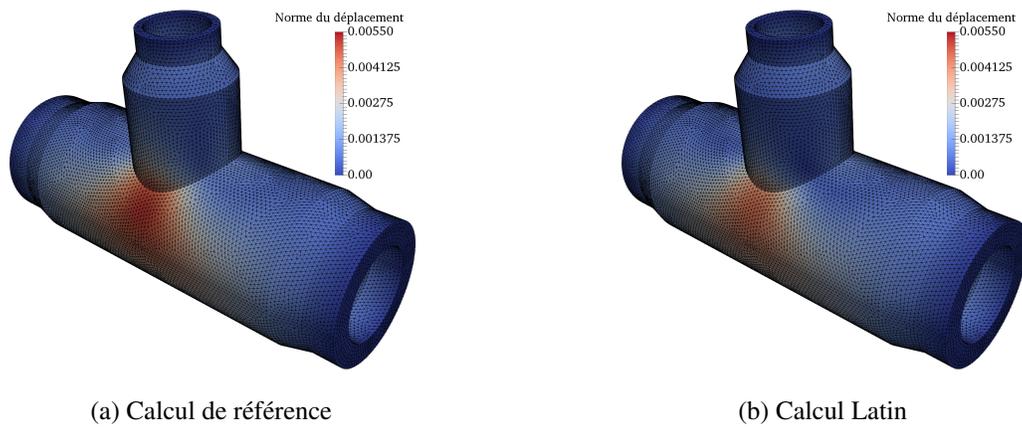


FIGURE 2.16 – Comparaison *Latin - référence* : norme du déplacement

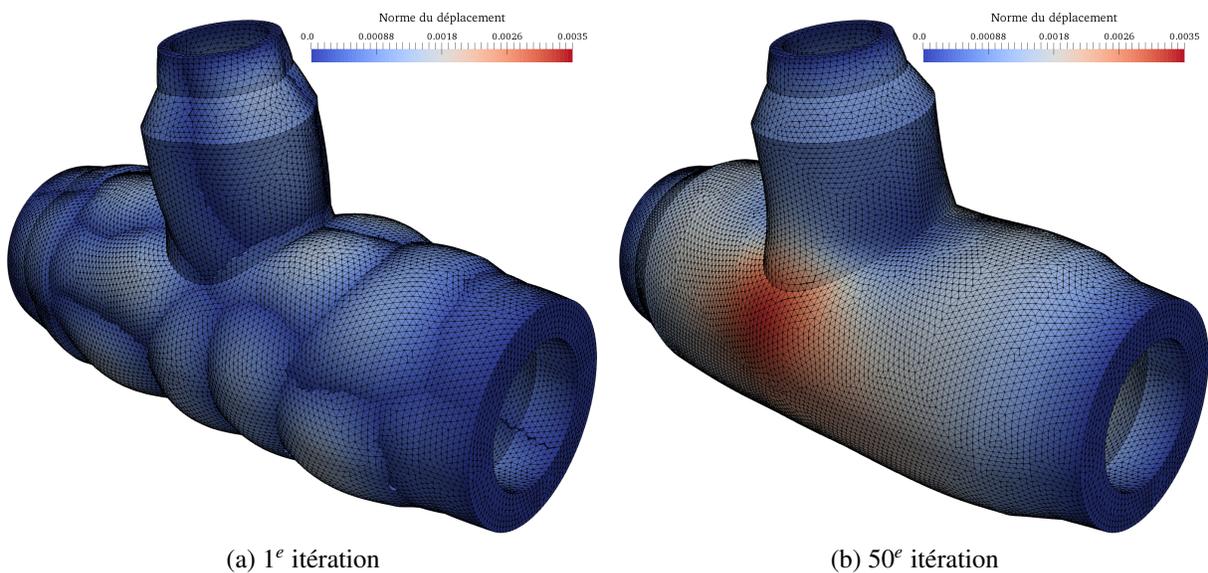


FIGURE 2.17 – Déformées à la 1^e et 50^e itérations

En effet les assemblages que nous étudierons ultérieurement seront constitués de pièces distinctes qui formeront chacune un sous-domaine. Les interfaces sont donc dans ce cas régulières et les problématiques énoncées précédemment à propos d'une sous-structuration automatique n'auront plus lieu d'être.

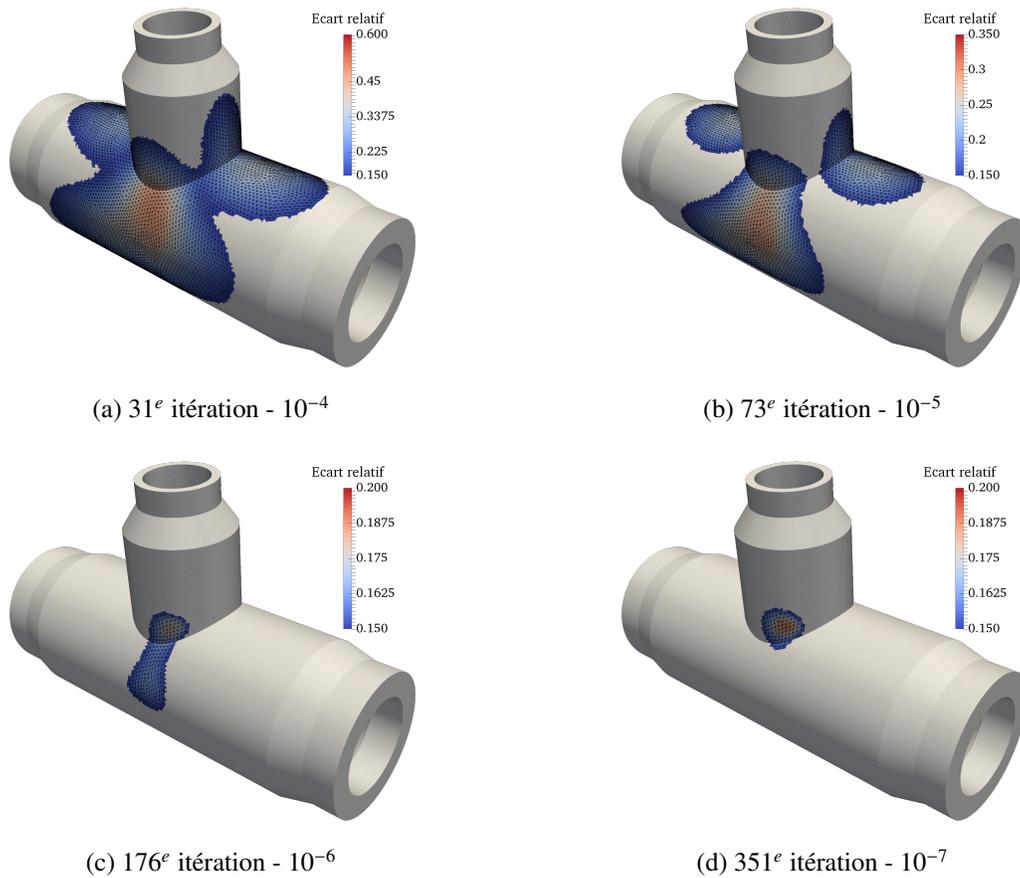


FIGURE 2.18 – Écart par rapport à la référence

4 Bilan

Dans ce chapitre a été présentée la méthode Latin appliquée à la décomposition de domaine (Section 2). L'idée est de distinguer les équations relatives aux sous-domaines des équations relatives aux interfaces. Cela permet de définir deux groupes de solutions partielles au problème dont il est plus facile de déterminer des membres : un espace dit *local* et un espace dit *linéaire*. Un schéma itératif à deux directions de recherche est mis en place pour passer successivement d'un espace de solutions partielles à l'autre et ainsi atteindre itérativement la solution du problème complet. L'algorithme se décompose alors en deux étapes : une *étape locale* sur les interfaces et une *étape linéaire* sur les sous-domaines. Le choix de ne privilégier ni les déplacements, ni les efforts au sein de la direction de recherche nous amène à gérer des conditions de Robin aux interfaces des sous-domaines. Le choix classique de direction de recherche écrite localement noeud à noeud conduit à un opérateur de direction de recherche qui n'est pas standard au sein des codes industriels (Section 2.4).

Nous proposons donc une nouvelle écriture de la direction de recherche via l'ajout d'une rigidité non-locale au niveau des interfaces (Section 3). L'ajout d'éléments aux interfaces permet

de transformer l'opérateur de direction de recherche classique en un opérateur de rigidité standard facilement calculable dans tout code industriel. Cette direction de "descente" qui permet de passer de l'espace local à l'espace linéaire est non-locale : des nœuds d'une même interface deviennent désormais couplés entre eux. Cela permet aussi de définir une nouvelle direction de "montée" (Section 3.1.4), pour passer de l'espace linéaire à l'espace local, qui peut donc devenir elle aussi non-locale. Les relations d'interface sont alors vérifiées par la résolution d'un problème de Neumann sur les semelles. L'objectif est de n'ajouter que très peu d'éléments pour éviter de trop augmenter la taille des problèmes sur les sous-domaines. Il est donc choisi d'ajouter une seule couche d'éléments dont le bord qui n'est pas en contact avec l'interface est encasté.

Une première implémentation non-intrusive a permis de se comparer à une implémentation standard de la méthode Latin (Section 3.2). Ce cas 1D vérifie en particulier une caractéristique classique de la Latin à savoir que la direction de recherche optimale ne dépend que de la structure globale et non du choix de sous-structuration. Nous nous sommes aussi intéressés à des indicateurs d'erreur différents de celui classiquement utilisé. Il a été présenté un indicateur quantifiant le saut de déplacement aux interfaces, un autre quantifiant le déséquilibre des efforts. Il a été vérifié avec ces indicateurs l'optimalité des directions de recherche obtenues par l'indicateur classique. Un calcul sur un cas 3D nous permet de comparer la solution Latin à une solution de référence déterminée directement par *code_aster* (Section 3.3). Nous avons pu mettre en évidence des problématiques liées aux cas d'interfaces non-régulières provoquées par une sous-structuration automatique. En dehors de zones localisées pour lesquelles la convergence est perturbée, la solution itérative tend vers la solution de référence. Nous ne nous sommes pas intéressés dans ce chapitre aux performances de la méthode Latin. Cela est l'objet de la section 3.1 du chapitre 5.

Gestion de l'étape linéaire : vision multi-échelle et accélération

Ce chapitre aborde la gestion de l'étape linéaire. Plus précisément nous présentons les ingrédients permettant l'extensibilité de la méthode Latin en décomposition de domaine à travers une réécriture de l'approche multi-échelle classique. Nous proposons également une évolution de l'étape de relaxation communément située à la fin de l'étape linéaire. L'idée étant de coupler le formalisme Latin avec un algorithme de Krylov afin d'accélérer la convergence de la méthode.

Sommaire

1	Introduction	70
2	Approche multi-échelle pour un équilibre global	72
2.1	Définition d'une base macroscopique en déplacement	72
2.1.1	Principe de la base macroscopique	72
2.1.2	La base macroscopique concrètement	73
2.2	Reécriture de l'étape linéaire	75
2.2.1	Obtention du problème	75
2.2.2	Approche classique de la stratégie multi-échelle	76
2.2.3	Nouvelle approche de la stratégie multi-échelle	77

	2.2.4	Illustration du couplage entre sous-domaines	78
2.3		Schéma de résolution du problème multi-échelle	80
	2.3.1	Algorithme à deux échelles	80
	2.3.2	Calcul de l'opérateur "macroscopique"	82
	2.3.3	Bilan sur l'algorithme de résolution	85
3		Approche multi-échelle pour une continuité globale	86
	3.1	Définition d'une base macroscopique en effort	86
	3.2	Définition de l'étape linéaire	86
		3.2.1 Ecriture du problème	86
		3.2.2 Obtention du problème en déplacement	87
	3.3	Schéma de résolution du problème en déplacement	88
4		Comparaison des deux approches - exemples numériques	89
	4.1	Étude de l'extensibilité	89
	4.2	Influence de la direction de recherche	94
	4.3	Influence du paramètre de relaxation	96
5		Accélération de type Krylov	97
	5.1	Principe de l'accélération de Krylov	97
		5.1.1 Quelques rappels sur les méthodes de Krylov	97
		5.1.2 Application à la méthode Latin	97
	5.2	Illustration	100
		5.2.1 Cas 2D	100
		5.2.2 Cas 3D	101
6		Bilan	103

1 Introduction

Il a été brièvement évoqué à la section 3.2 une dépendance de la convergence à la sous-structuration. En reprenant l'exemple 1D de poutre en traction nous illustrons cette dépendance à la figure 3.1. Plus le nombre de sous-domaines augmente, plus il est requis d'itérations pour faire transiter les informations des conditions limites à travers la structure. Cette méthode n'est donc pas extensible et il est nécessaire de rajouter un ingrédient supplémentaire pour qu'elle devienne indépendante de la sous-structuration.

Les premiers travaux sur l'extensibilité de la méthode Latin pour la décomposition de domaine datent de [Dureisseix, 1997]. L'idée est de s'inspirer des méthodes multi-grilles [Parsons and Hall, 1990] et de résoudre le problème à deux échelles. Une échelle globale est liée aux phénomènes à grande longueur d'onde et à la structure globale. L'autre échelle plus fine permet de capter des phénomènes locaux. Le problème à l'échelle globale est résolu sur un maillage grossier par une méthode éléments finis standard. Le problème à l'échelle fine est quant à lui résolu par une technique de décomposition de domaine.

Une vision par homogénéisation est proposée par [Ladevèze and Dureisseix, 1999] pour laquelle une distinction micro / macro est effectuée sur les grandeurs de l'étape linéaire. Il est défini

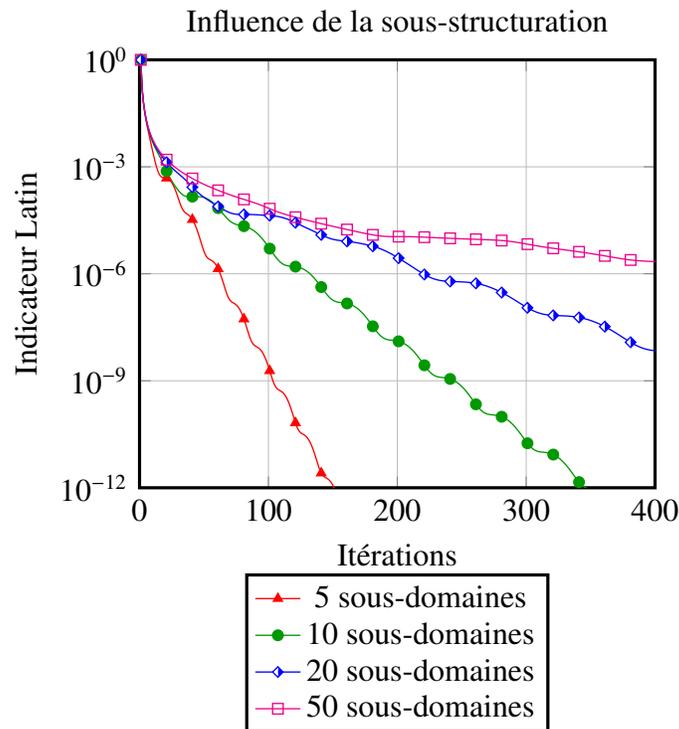


FIGURE 3.1 – Illustration de la dépendance à la sous-structuration

une base macroscopique des déplacements et des efforts d'interface. Une direction de recherche globale spécifique est appliquée aux quantités macro alors qu'une direction de recherche classique est appliquée aux grandeurs micro. [Loiseau, 2001, Ladevèze et al., 2001] ont mis en oeuvre cette approche pour le cas de structures hétérogènes. Dans ces travaux une approche visant à vérifier globalement l'équilibre des efforts d'interface lors de l'étape linéaire a été proposée. Une approche pour vérifier une continuité globale des champs de déplacement d'interface a aussi été présentée.

Dans [Ladevèze and Nouy, 2003, Nouy, 2003] l'équilibre macroscopique des efforts à l'étape linéaire est imposé par un multiplicateur de Lagrange. Une troisième échelle pour la résolution d'un problème *super-macro* a été introduite. Cette approche est utilisée par [Violeau, 2007] pour le cas de structures composites. Cette idée de mettre en place une échelle supplémentaire a été ensuite développée par [Kerfriden, 2008, Kerfriden et al., 2009] pour l'étude du délaminage dans les structures composites. Un trop grand nombre de sous-domaines conduit à un problème macroscopique de grande taille et il est proposé d'introduire un solveur BDD pour le résoudre.

Dans le cadre de ce manuscrit nous nous restreignons à une méthode à deux échelles. Nous proposons une réécriture de l'approche multi-échelle proposée par [Ladevèze and Nouy, 2003]. Nous n'effectuons pas de distinction micro / macro de la relation de direction de recherche et montrons aussi qu'il n'est pas nécessaire de définir *a priori* de base macroscopique en effort. Nous reprenons aussi l'idée d'imposer une continuité globale des déplacements d'interface lors de l'étape linéaire. Contrairement à [Loiseau, 2001] nous prenons en compte cette condition par multiplicateur.

Lors de l'étape linéaire telle que présentée à la section 3.1.4 les grandeurs d'interface relatives

aux sous-domaines ne vérifient pas les relations d'équilibre des efforts et de continuité des déplacements. L'idée de l'approche multi-échelle est de modifier l'étape linéaire afin d'imposer une vérification globale des lois de comportement d'interface.

2 Approche multi-échelle pour un équilibre global

2.1 Définition d'une base macroscopique en déplacement

2.1.1 Principe de la base macroscopique

Dans la version multi-échelle de la méthode Latin, il est nécessaire de définir une base macroscopique de déplacement d'interface par rapport à laquelle les efforts seront ensuite équilibrés. Ainsi pour chaque interface on définit une base de déplacement macroscopique. La contrainte imposée, liée à ces modes macroscopiques, permet de transmettre les informations à grande longueur d'ondes à travers la structure. Afin de satisfaire le principe de Saint-Venant, la base doit contenir la trace des mouvements de corps rigide. La pratique usuelle de la méthode Latin est de considérer ces modes rigides par interfaces (Figure 3.2 et 3.3) et d'incorporer à cette base des modes d'extension (Figure 3.4 et Figure 3.5). En considérant une interface $\Gamma_{EE'}$, la base macroscopique de déplacement est notée $\mathbb{W}_{EE'}$. Nous considérons \mathbb{W} la base macroscopique concaténée.

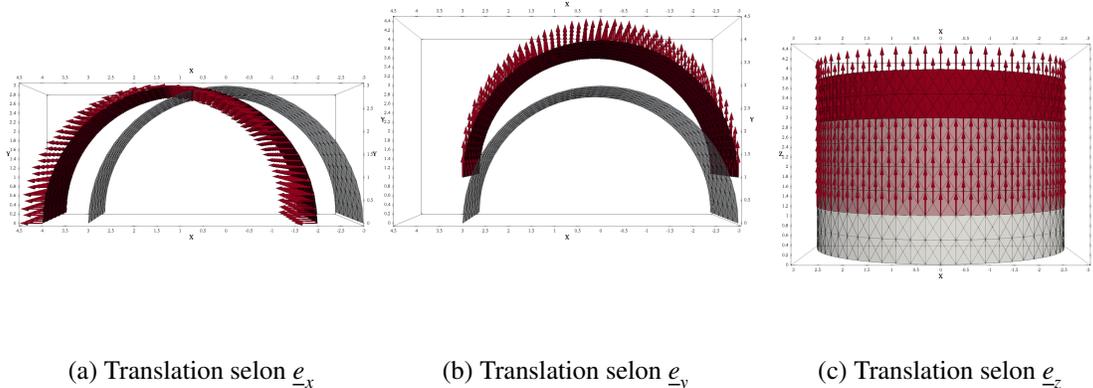


FIGURE 3.2 – Translations

Remarque 3.0

De nombreux choix de base macroscopique de déplacement existent. [Saavedra et al., 2017] se sont intéressés à l'utilisation de modes d'ordre supérieur. Il a été montré en particulier que cet enrichissement n'améliore pas significativement les performances. Il suffit donc de conserver des modes d'ordre 1 comme les mouvements de corps rigides ainsi que des déformations simples.

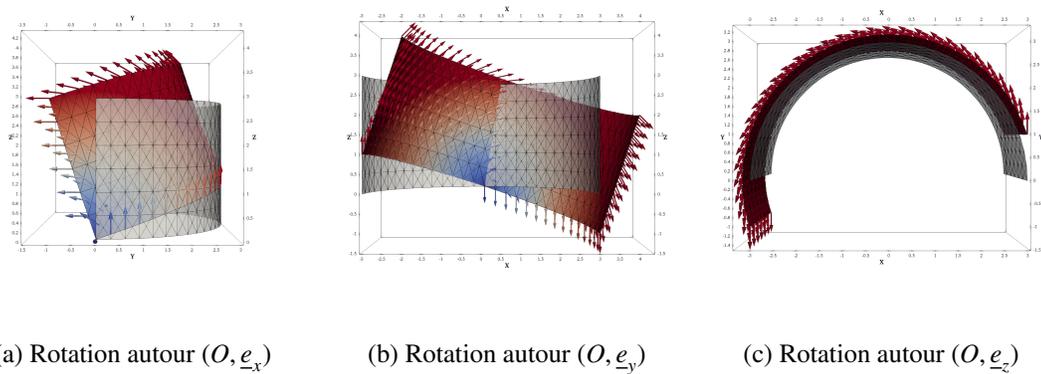


FIGURE 3.3 – Rotations

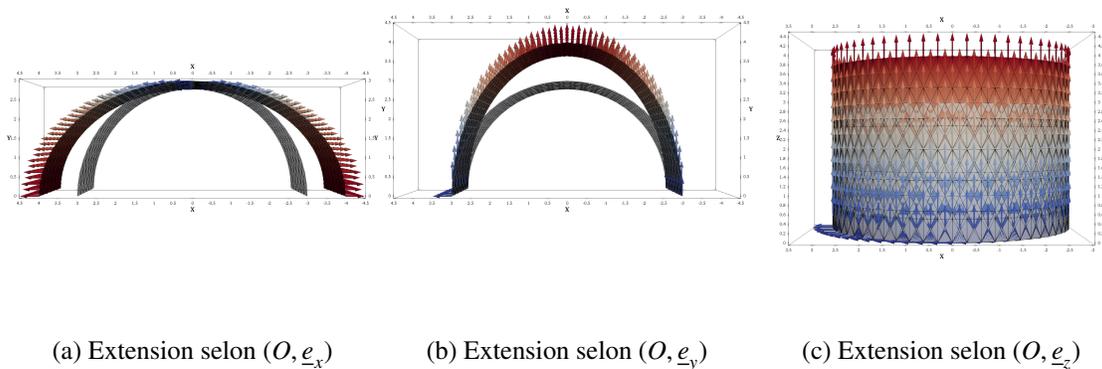
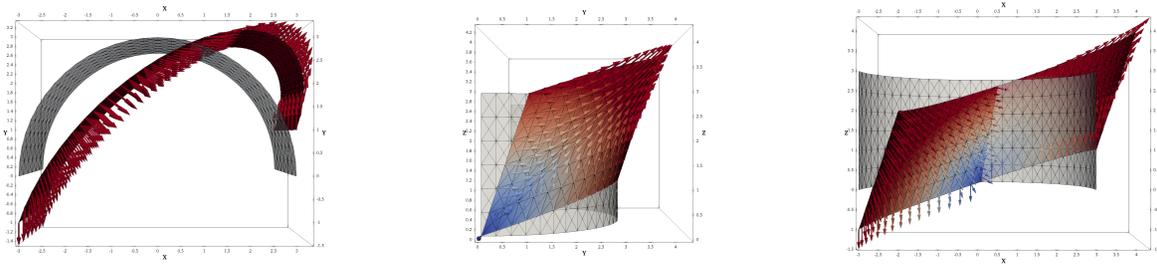


FIGURE 3.4 – Extensions

2.1.2 La base macroscopique concrètement

Quel que soit le type d'interface, régulière ou non, les modes de corps rigides sont faciles à implémenter. Cependant dans le cas d'interfaces non planes ou non régulières, il est plus complexe de faire ressortir des modes de déformation. [Violeau, 2007] propose une écriture générale pour les modes macroscopiques. Cette formulation fait intervenir les directions principales d'inertie et permet d'écrire les modes de déformation suivant les axes principaux d'inertie. Cependant dans le cadre non-intrusif, il s'avère que le calcul des directions principales d'inertie n'est pas accessible facilement pour les éléments d'interface (fonction `POST_ELEM` de `code_aster`). L'intérêt d'une telle base dans un repère principal d'inertie n'est pas flagrant donc nous proposons de rester dans un cadre général pour lequel les déformations sont écrites dans la base canonique $(O, \underline{e}_x, \underline{e}_y, \underline{e}_z)$. Ainsi en considérant un point M de l'interface Γ , nous choisissons les modes macroscopiques suivant :



(a) Cisaillement dans le plan $(O, \underline{e}_x, \underline{e}_y)$ (b) Cisaillement dans le plan $(O, \underline{e}_y, \underline{e}_z)$ (c) Cisaillement dans le plan $(O, \underline{e}_x, \underline{e}_z)$

FIGURE 3.5 – Cisaillements

— Les modes rigides de translation et rotation :

$$\begin{aligned}
 \underline{U}_x = \underline{e}_x \quad \text{Translation suivant } \underline{e}_x & \quad \underline{R}_x = \frac{\underline{e}_x \wedge \underline{OM}}{\max_{M \in \Gamma} \{ \|\underline{e}_x \wedge \underline{OM}\| \}} \text{Rotation autour de } \underline{e}_x \\
 \underline{U}_y = \underline{e}_y \quad \text{Translation suivant } \underline{e}_y & \quad \underline{R}_y = \frac{\underline{e}_y \wedge \underline{OM}}{\max_{M \in \Gamma} \{ \|\underline{e}_y \wedge \underline{OM}\| \}} \text{Rotation autour de } \underline{e}_y \quad (3.1) \\
 \underline{U}_z = \underline{e}_z \quad \text{Translation suivant } \underline{e}_z & \quad \underline{R}_z = \frac{\underline{e}_z \wedge \underline{OM}}{\max_{M \in \Gamma} \{ \|\underline{e}_z \wedge \underline{OM}\| \}} \text{Rotation autour de } \underline{e}_z
 \end{aligned}$$

— Les modes d'extension et de cisaillement :

$$\begin{aligned}
 \underline{u}_x = \frac{(\underline{e}_x \ 0 \ 0) \cdot \underline{OM}}{\max_{M \in \Gamma} \{ \|\underline{e}_x \ 0 \ 0\} \cdot \underline{OM}\|} \quad \text{Extension suivant } \underline{e}_x & \quad \underline{u}_{xy} = \frac{(\underline{e}_y \ \underline{e}_x \ 0) \cdot \underline{OM}}{\max_{M \in \Gamma} \{ \|\underline{e}_y \ \underline{e}_x \ 0\} \cdot \underline{OM}\|} \quad \text{Cisaillement suivant } (\underline{e}_x, \underline{e}_y) \\
 \underline{u}_y = \frac{(0 \ \underline{e}_y \ 0) \cdot \underline{OM}}{\max_{M \in \Gamma} \{ \|(0 \ \underline{e}_y \ 0) \cdot \underline{OM}\|} & \quad \underline{u}_{xz} = \frac{(\underline{e}_z \ 0 \ \underline{e}_x) \cdot \underline{OM}}{\max_{M \in \Gamma} \{ \|\underline{e}_z \ 0 \ \underline{e}_x\} \cdot \underline{OM}\|} \quad \text{Cisaillement suivant } (\underline{e}_x, \underline{e}_z) \quad (3.2) \\
 \underline{u}_z = \frac{(0 \ 0 \ \underline{e}_z) \cdot \underline{OM}}{\max_{M \in \Gamma} \{ \|(0 \ 0 \ \underline{e}_z) \cdot \underline{OM}\|} & \quad \underline{u}_{yz} = \frac{(0 \ \underline{e}_z \ \underline{e}_y) \cdot \underline{OM}}{\max_{M \in \Gamma} \{ \|(0 \ \underline{e}_z \ \underline{e}_y) \cdot \underline{OM}\|} \quad \text{Cisaillement suivant } (\underline{e}_y, \underline{e}_z)
 \end{aligned}$$

⚠ Remarque 3.1

Pour certaines configurations, il est possible que des modes ainsi définis soient égaux. Par exemple en considérant une interface orthogonale à \underline{e}_x , le mode de translation et le mode d'extension sont identiques. Ainsi une élimination des modes colinéaires ou un traitement particulier des opéra-

teurs macroscopiques doivent être mis en place.

2.2 Réécriture de l'étape linéaire

2.2.1 Obtention du problème

La condition d'équilibre global des efforts d'interface consiste à imposer que le travail des inter-efforts \mathbf{AF} par rapport aux modes macroscopiques de déplacement \mathbb{W} est nul :

$$\mathbb{W}^T \mathbf{AF} = 0 \quad (3.3)$$

La prise en compte de cette nouvelle relation impose de reconsidérer la relation de direction de recherche. Elle est désormais définie globalement sur toutes les interfaces et non plus localement interface par interface. La direction de recherche est donc formulée comme un problème de minimisation sous contrainte d'égalité :

$$\min_{\mathbb{W}^T \mathbf{AF}=0} \left\{ \frac{1}{2} (\mathbf{F} - \widehat{\mathbf{F}})^T \mathbf{k}^{-1} (\mathbf{F} - \widehat{\mathbf{F}}) + (\mathbf{F} - \widehat{\mathbf{F}})^T (\mathbf{W} - \widehat{\mathbf{W}}) \right\} \quad (3.4)$$

La contrainte est prise en compte par la définition du lagrangien $\mathcal{L}(\mathbf{F}, \alpha)$. α est le multiplicateur de Lagrange associé à la contrainte $\mathbb{W}^T \mathbf{AF} = 0$.

$$\mathcal{L}(\mathbf{F}, \alpha) = \frac{1}{2} (\mathbf{F} - \widehat{\mathbf{F}})^T \mathbf{k}^{-1} (\mathbf{F} - \widehat{\mathbf{F}}) + (\mathbf{F} - \widehat{\mathbf{F}})^T (\mathbf{W} - \widehat{\mathbf{W}}) + \alpha^T \mathbb{W}^T \mathbf{AF} \quad (3.5)$$

La minimisation du lagrangien par rapport aux efforts d'interface \mathbf{F} conduit à la nouvelle écriture de la direction de recherche :

$$(\mathbf{F} - \widehat{\mathbf{F}}) + \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) + \mathbf{k}^- \mathbf{A}^T \mathbb{W} \alpha = 0 \quad (3.6)$$

Cette direction de recherche couple les parties macroscopiques des efforts d'interface de chaque sous-domaine voisin. La minimisation du lagrangien par rapport au multiplicateur α redonne la contrainte d'équilibre global des efforts d'interface.

Remarque 3.2

Dans la littérature l'approche multi-échelle est écrite avec une formulation continue [Ladevèze and Nouy, 2003, Violeau, 2007]. Le multiplicateur est écrit \tilde{W} et correspond à $\mathbf{A}^T \mathbb{W} \alpha$.

Le problème à résoudre à l'étape linéaire devient avec l'approche multi-échelle :

$$\left. \begin{array}{l} \text{Trouver } (\mathbf{W}, \mathbf{F}) \text{ solution de :} \\ \left\{ \begin{array}{l} \mathbf{Ku} = \mathbf{f}_d + \mathbf{N}^T \mathbf{F} \\ \mathbf{W} = \mathbf{Nu} \\ \mathbb{W}^T \mathbf{AF} = 0 \\ (\mathbf{F} - \widehat{\mathbf{F}}) + \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) + \mathbf{k}^- \mathbf{A}^T \mathbb{W} \alpha = 0 \end{array} \right. \end{array} \right\} \quad (3.7)$$

2.2.2 Approche classique de la stratégie multi-échelle

Classiquement la résolution du problème 3.7 se fait en quatre étapes dont le schéma est illustré à la figure 3.6. La relation de trace entre \mathbf{W} et \mathbf{u} est omise volontairement. Elle est implicitement utilisée pour passer de \mathbf{u} à \mathbf{W} ou inversement. En notant $\tilde{\mathbf{k}} = \mathbf{N}^T \mathbf{k}^- \mathbf{N}$ pour ne pas surcharger les équations, le schéma est décomposé comme suit :

1. La relation de direction de recherche est injectée au sein de l'équation d'équilibre des sous-domaines : Cela permet d'obtenir l'expression du déplacement \mathbf{u} en fonction du multiplicateur α : $\mathbf{u} = g_u(\alpha)$:

$$\mathbf{u} = (\mathbf{K} + \tilde{\mathbf{k}})^{-1} [\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}} - \mathbf{k}^- \mathbf{A}^T \mathbb{W} \alpha)] \quad (3.8)$$

2. \mathbf{W} est obtenu par l'opérateur de trace \mathbf{N} , il est ensuite injecté dans la relation de direction de recherche permettant d'obtenir la relation entre \mathbf{F} et α : $\mathbf{F} = g_F(\alpha)$.

$$\mathbf{F} = (\mathbf{I} - \mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T) [\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}} - \mathbf{k}^- \mathbf{A}^T \mathbb{W} \alpha] - \mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{f}_d \quad (3.9)$$

3. L'expression de \mathbf{F} est insérée dans la relation d'équilibre global des efforts d'interface. Cela permet de faire disparaître \mathbf{F} de la relation et d'obtenir l'expression finale de α .

$$\alpha = (\mathbb{W}^T \mathbf{A} (\mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}}) \mathbf{N}^T - \mathbf{I}) \mathbf{k}^- \mathbf{A}^T \mathbb{W})^{-1} \mathbb{W}^T \mathbf{A} ((\mathbf{K} + \tilde{\mathbf{k}}) \mathbf{N}^T - \mathbf{I}) [\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}] \quad (3.10)$$

4. Le multiplicateur α connu, le déplacement \mathbf{u} est déterminé via la relation $\mathbf{u} = g_u(\alpha)$.

$$\mathbf{u} = (\mathbf{K} + \tilde{\mathbf{k}})^{-1} [\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) - \mathbf{N}^T \mathbf{k}^- \mathbf{A}^T \mathbb{W} (\mathbb{W}^T \mathbf{A} (\mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}}) \mathbf{N}^T - \mathbf{I}) \mathbf{k}^- \mathbf{A}^T \mathbb{W})^{-1} \dots \dots \mathbb{W}^T \mathbf{A} \mathbf{k}^- (\mathbf{N} - \widehat{\mathbf{W}}) (\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}))] \quad (3.11)$$

Une fois le déplacement \mathbf{u} connu sur les sous-domaines, les grandeurs d'interface \mathbf{W} et \mathbf{F} sont déterminées respectivement par l'opérateur de trace \mathbf{N} et la relation de direction de recherche.

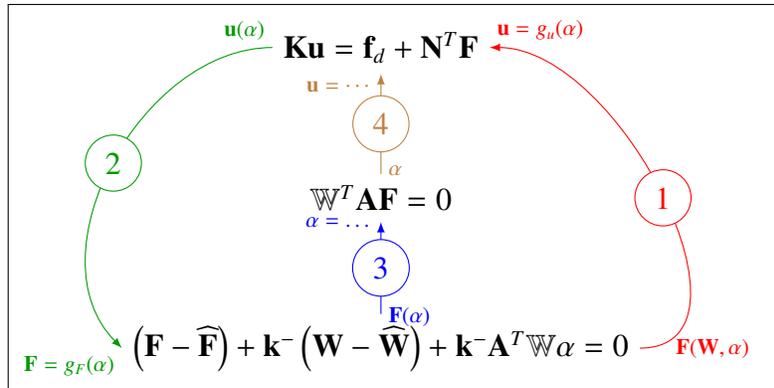


FIGURE 3.6 – Séquence de résolution du problème classique

2.2.3 Nouvelle approche de la stratégie multi-échelle

Nous proposons de modifier la séquence présentée précédemment. Cette fois ci le schéma se décompose en trois étapes :

1. La relation de direction de recherche est injectée dans la relation d'équilibre global :

$$\underbrace{\mathbb{W}^T \mathbf{A} \mathbf{F}}_{=0 \text{ (équilibre macro)}} - \mathbb{W}^T \underbrace{\mathbf{A} \widehat{\mathbf{F}}}_{=0 \text{ (équilibre local)}} + \mathbb{W}^T \mathbf{A} \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) + \mathbb{W}^T \mathbf{A} \mathbf{k}^- \mathbf{A}^T \mathbb{W} \alpha = 0 \quad (3.12)$$

Cela permet d'exprimer le multiplicateur α en fonction de l'inconnue de déplacement \mathbf{W} :
 $\alpha = f_\alpha(\mathbf{W}) :$

$$\alpha = \left(\mathbb{W}^T \mathbf{A} \mathbf{k}^- \mathbf{A}^T \mathbb{W} \right)^{-1} \left[\mathbb{W}^T \mathbf{A} \mathbf{k}^- (\widehat{\mathbf{W}} - \mathbf{W}) \right] \quad (3.13)$$

2. L'expression du multiplicateur est ensuite re-injectée dans la relation de direction de recherche pour obtenir une relation entre \mathbf{F} et \mathbf{W} : $\mathbf{F} = f_F(\mathbf{W}) :$

$$\mathbf{F} = \mathbf{k}^- \left[\mathbf{I} - \mathbf{A}^T \mathbb{W} \left(\mathbb{W}^T \mathbf{A} \mathbf{k}^- \mathbf{A}^T \mathbb{W} \right)^{-1} \mathbb{W}^T \mathbf{A} \mathbf{k}^- \right] (\widehat{\mathbf{W}} - \mathbf{W}) + \widehat{\mathbf{F}} \quad (3.14)$$

3. La dernière étape consiste à insérer cette relation entre les inconnues d'effort et de déplacement dans la relation d'équilibre des sous-domaines pour obtenir le problème en déplacement à résoudre :

$$\boxed{\left[\mathbf{K} + \mathbf{N}^T \mathbf{k}^- (\mathbf{I} - \mathbf{P}_M) \mathbf{N} \right] \mathbf{u} = \mathbf{f}_d + \mathbf{N}^T \left[\widehat{\mathbf{F}} + \mathbf{k}^- (\mathbf{I} - \mathbf{P}_M) \widehat{\mathbf{W}} \right]} \quad (3.15)$$

Avec $\mathbf{P}_M = \mathbf{A}^T \mathbb{W} \left(\mathbb{W}^T \mathbf{A} \mathbf{k}^- \mathbf{A}^T \mathbb{W} \right)^{-1} \mathbb{W}^T \mathbf{A} \mathbf{k}^-$

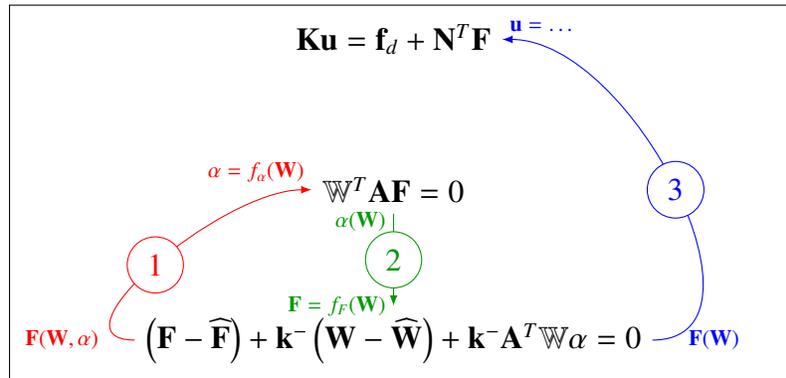


FIGURE 3.7 – Séquence de résolution du problème revisité

Nous avons pu mettre en évidence à l'équation 3.15 un projecteur \mathbf{P}_M que nous qualifierons de *macroscopique*. Plusieurs observations peuvent déjà être faites quant à la forme de ce projecteur :

- l'expression du projecteur ne fait apparaître que les paramètres de direction de recherche \mathbf{k}^- ainsi que les bases macroscopiques de déplacement. La rigidité des sous-domaines n'intervient pas.

- l'approche multi-échelle peut être vue comme un choix particulier de direction de recherche. En effet dans l'écriture on reconnaît la direction de recherche mono-échelle assouplie par le projecteur \mathbf{P}_M .
- le projecteur \mathbf{P}_M couple les sous-domaines voisins entre eux, une illustration sur un cas simple est montrée à la section 2.2.4.

A partir de \mathbf{P}_M il est possible de définir un projecteur *microscopique* $\mathbf{P}_m = \mathbf{I} - \mathbf{P}_M$ qui filtre la partie macroscopique du déplacement. La forme du projecteur microscopique nous permet d'en déduire son noyau ainsi que son image :

$$\begin{aligned} \ker \mathbf{P}_m &= \text{Im}(\mathbf{A}^T \mathbb{W}) \\ \text{Im} \mathbf{P}_m &= \ker(\mathbb{W}^T \mathbf{A} \mathbf{k}^-) \end{aligned} \quad (3.16)$$

⚠ Remarque 3.3

Nous n'avons pas défini a priori de base macroscopique d'effort. Cependant il est possible d'en déduire une à partir de celle en déplacement. En effet en choisissant un espace macroscopique $\mathcal{F}^M = \text{Vect}(\mathbf{k}^- \mathbf{A}^T \mathbb{W})$ on vérifie la propriété selon laquelle les efforts macroscopiques ne travaillent pas suivant les déplacements microscopiques. Cette base en effort n'est identique entre deux voisins que s'ils utilisent le même opérateur de direction descente \mathbf{k}^- .

L'image et le noyau du projecteur \mathbf{P}_m permettent de déduire une forme du déplacement d'interface \mathbf{W} . Il peut se décomposer comme une partie micro qui ne travaille pas avec les efforts macro, ainsi qu'une partie macro filtrée par le projecteur microscopique.

$$\mathbf{W} = \mathbf{A}^T \mathbb{W} \beta + \mathbf{W}_m \quad (3.17)$$

avec $\mathbf{A}^T \mathbb{W} \beta$ qui représente la partie macro, et \mathbf{W}_m la partie micro qui vérifie $\mathbb{W}^T \mathbf{A} \mathbf{k}^- \mathbf{W}_m = 0$

2.2.4 Illustration du couplage entre sous-domaines

Afin d'illustrer le couplage entre les sous-domaines, nous considérons un problème simple à trois sous-domaines et deux interfaces

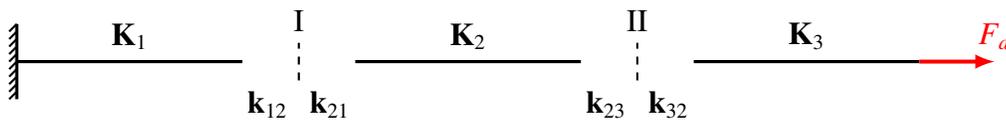


FIGURE 3.8 – Exemple pour le couplage entre sous-domaines

L'opérateur de rigidité mono-échelle s'écrit :

$$\mathbf{K}_{mono} = \begin{pmatrix}
 \begin{array}{|c|c|c|}
 \hline
 \mathbf{K}_1^i & \mathbf{K}_{12}^i & \\
 \hline
 \mathbf{K}_{12}^i & \mathbf{K}_{12} + \mathbf{k}_{12} & 0 \\
 \hline
 0 & \mathbf{K}_{21}^i + \mathbf{k}_{12} & \mathbf{K}_{21}^i & \mathbf{K}_{21}^{23} \\
 \hline
 \mathbf{K}_{21}^i & \mathbf{K}_2^i & \mathbf{K}_{23}^i & \\
 \hline
 \mathbf{K}_{21}^{23} & \mathbf{K}_{23}^i & \mathbf{K}_{23} + \mathbf{k}_{23} & 0 \\
 \hline
 0 & 0 & \mathbf{K}_{32} + \mathbf{k}_{32} & \mathbf{K}_{32}^i \\
 \hline
 \mathbf{K}_{32}^i & \mathbf{K}_3^i & & \\
 \hline
 \end{array}
 \end{pmatrix} \quad (3.18)$$

Sous-domaine 1

Sous-domaine 2

Sous-domaine 3

Decouplage des sous-domaines

Il est aisément observé que dans le cas mono-échelle les sous-domaines voisins (1/2 et 2/3) sont découplés. Nous allons désormais illustrer l'influence du projecteur \mathbf{P}_M sur cet opérateur. Nous rappelons tout d'abord quelques opérateurs adaptés au problème étudié.

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & \mathbf{I} \end{pmatrix}, \quad \mathbb{W} = \begin{pmatrix} \mathbb{W}_{12} \\ \mathbb{W}_{23} \end{pmatrix}, \quad \mathbf{k} = \begin{pmatrix} \mathbf{k}_{12} & & & \\ & \mathbf{k}_{21} & & \\ & & \mathbf{k}_{23} & \\ & & & \mathbf{k}_{32} \end{pmatrix} \quad (3.19)$$

Ainsi en notant $\gamma = \begin{pmatrix} \gamma_{11} & 0 \\ 0 & \gamma_{22} \end{pmatrix}$ tel que :

$$\gamma = (\mathbb{W}^T \mathbf{A} \mathbf{k} \mathbf{A}^T \mathbb{W})^{-1} = \begin{pmatrix} (\mathbb{W}_{12}^T (\mathbf{k}_{12} + \mathbf{k}_{21}) \mathbb{W}_{12})^{-1} & 0 \\ 0 & (\mathbb{W}_{23}^T (\mathbf{k}_{23} + \mathbf{k}_{32}) \mathbb{W}_{23})^{-1} \end{pmatrix} \quad (3.20)$$

l'expression de \mathbf{P}_M :

$$\mathbf{P}_M = \begin{pmatrix} \lambda_{11} & \lambda_{12} & & \\ \lambda_{21} & \lambda_{22} & & \\ & & \lambda_{33} & \lambda_{34} \\ & & \lambda_{43} & \lambda_{44} \end{pmatrix} = \begin{pmatrix} \mathbb{W}_{12}^T \gamma_{11} \mathbb{W}_{12}^T \mathbf{k}_{12} & \mathbb{W}_{12}^T \gamma_{11} \mathbb{W}_{12}^T \mathbf{k}_{21} & & \\ \mathbb{W}_{12}^T \gamma_{11} \mathbb{W}_{12}^T \mathbf{k}_{12} & \mathbb{W}_{12}^T \gamma_{11} \mathbb{W}_{12}^T \mathbf{k}_{21} & & \\ & & \mathbb{W}_{23}^T \gamma_{22} \mathbb{W}_{23}^T \mathbf{k}_{23} & \mathbb{W}_{23}^T \gamma_{22} \mathbb{W}_{23}^T \mathbf{k}_{32} \\ & & \mathbb{W}_{23}^T \gamma_{22} \mathbb{W}_{23}^T \mathbf{k}_{23} & \mathbb{W}_{23}^T \gamma_{22} \mathbb{W}_{23}^T \mathbf{k}_{32} \end{pmatrix} \quad (3.21)$$

et l'opérateur de rigidité global devient dans le cas multi-échelle :

Avec une telle écriture de l'inverse de l'opérateur de rigidité global, il est possible d'exprimer le déplacement sous la forme :

$$\begin{aligned} \mathbf{u} = & (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) - \dots \\ & \dots (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{k}^- \mathbf{A}^T \mathbb{W} \left(\mathbb{W}^T \mathbf{A} (\mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T - \mathbf{I}) \mathbf{k}^- \mathbf{A}^T \mathbb{W} \right)^{-1} \dots \\ & \dots \mathbb{W} \mathbf{A} \mathbf{k}^- \left[\widehat{\mathbf{W}} - \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) \right] \end{aligned} \quad (3.26)$$

Remarque 3.4

Cette écriture est équivalente à celle obtenue par l'approche classique. Nous avons seulement modifié le schéma pour atteindre le résultat. Cela nous a permis d'en déduire quelques propriétés algébriques sur les espaces micro et macro des grandeurs d'interface.

Suivant l'écriture du déplacement à l'équation 3.26, une stratégie à deux échelles est mise en place pour laquelle sur chaque sous-domaine en parallèle la solution au problème mono-échelle est calculée :

$$\mathbf{u}_1 = (\mathbf{K} + \tilde{\mathbf{k}})^{-1} (\mathbf{f}_d + \mathbf{N}^T [\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}]) \quad (3.27)$$

On reconnaît dans l'expression du second terme de l'équation 3.26 le premier terme $(\mathbf{K} + \tilde{\mathbf{k}})^{-1} (\mathbf{f}_d + \mathbf{N}^T [\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}])$, ainsi la solution déterminée lors de la première étape est alors utilisée pour le calcul du second terme :

$$\mathbf{u}_2 = -(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{k}^- \mathbf{A}^T \mathbb{W} \mathbf{K}_M^{-1} \mathbb{W}^T \mathbf{A} \mathbf{k}^- [\mathbf{W}_1 - \widehat{\mathbf{W}}] \quad (3.28)$$

avec $\mathbf{W}_1 = \mathbf{N} \mathbf{u}_1$ qui correspond aux traces des déplacements induits par la solution du problème mono-échelle et $\mathbf{K}_M = \left(\mathbb{W}^T \mathbf{A} (\mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T - \mathbf{I}) \mathbf{k}^- \mathbf{A}^T \mathbb{W} \right)$ un opérateur "macroscopique" qui couple les sous-domaines voisins entre eux. Cet opérateur est constant au cours des itérations et il est assemblé et factorisé au début de l'algorithme.

La solution en déplacement \mathbf{u} est finalement la somme des deux solutions \mathbf{u}_1 et \mathbf{u}_2 déterminées via ces deux étapes.

$$\mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2 \quad (3.29)$$

Remarque 3.5

Par abus de langage, \mathbf{u}_1 et \mathbf{u}_2 sont classiquement appelées *déplacement micro* et *déplacement macro*. Cependant ces deux parties du champs de déplacement ne respectent pas les propriétés algébriques que nous avons présentées à la section 2.2.3. Par exemple $\mathbf{P}_m \mathbf{N} \mathbf{u}_2 \neq 0$, c'est-à-dire que \mathbf{u}_2 n'appartient pas à l'espace des déplacements macroscopiques.

Le multiplicateur s'exprime par :

$$\alpha = \mathbf{K}_M^{-1} \mathbb{W}^T \mathbf{A} (\mathbf{W}_1 - \widehat{\mathbf{W}}) \quad (3.30)$$

qui est déterminé lors du calcul de \mathbf{u}_2 . Ainsi la détermination des efforts d'interface \mathbf{F} avec la relation de direction de recherche ne nécessite pas de communication supplémentaire entre les sous-domaines.

Classiquement le problème qui consiste à déterminer \mathbf{u}_1 est appelé *problème microscopique*. Le *problème macroscopique* consiste à déterminer le multiplicateur $\alpha = \mathbf{K}_M^{-1} \mathbb{W}^T \mathbf{A} \mathbf{k}^{-1} [\mathbf{W}_1 - \widehat{\mathbf{W}}]$ dans l'expression de \mathbf{u}_2 . L'opérateur $(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{k}^{-1} \mathbf{A}^T \mathbb{W}$ étant déjà déterminé lors de la factorisation de l'opérateur macroscopique, la résolution du second *problème microscopique* ne nécessite pas une nouvelle résolution sur les sous-domaines pour déterminer $(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{k}^{-1} \mathbf{A}^T \mathbb{W} \alpha$.

2.3.2 Calcul de l'opérateur "macroscopique"

Nous détaillons ici comment obtenir l'opérateur macroscopique \mathbf{K}_M . Pour cela nous nous plaçons dans un cas simple de trois sous-domaines reliés par deux interfaces comme l'exemple de la section 3.1.1 que nous rappelons à la figure 3.9.

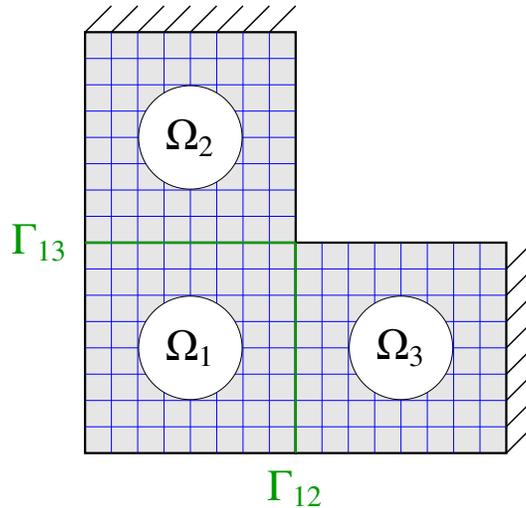


FIGURE 3.9 – Exemple pour illustrer la construction de \mathbf{K}_M

Nous considérons une base macroscopique de déplacement constituée de deux modes macro par interface : deux translations par exemple.

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbb{W} = \begin{bmatrix} \mathbb{W}_{12}^1 & \mathbb{W}_{12}^2 \\ \mathbb{W}_{13}^1 & \mathbb{W}_{13}^2 \end{bmatrix} \quad (3.31)$$

Dans ce cas $\mathbf{A}^T \mathbb{W}$ s'écrit :

$$\mathbf{A}^T \mathbb{W} = \begin{bmatrix} \begin{bmatrix} \mathbb{W}_{12}^1 & \mathbb{W}_{12}^2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \mathbb{W}_{13}^1 & \mathbb{W}_{13}^2 \end{bmatrix} \\ \begin{bmatrix} \mathbb{W}_{12}^1 & \mathbb{W}_{12}^2 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \mathbb{W}_{13}^1 & \mathbb{W}_{13}^2 \end{bmatrix} \end{bmatrix} \quad (3.32)$$

Appliquer l'opérateur de direction de recherche \mathbf{k} revient à écrire :

$$\mathbf{kA}^T \mathbb{W} = \begin{pmatrix} \boxed{\mathbf{k}_{12} \mathbb{W}_{12}^1 \quad \mathbf{k}_{12} \mathbb{W}_{12}^2} & & & \\ & \boxed{\mathbf{k}_{13} \mathbb{W}_{13}^1 \quad \mathbf{k}_{13} \mathbb{W}_{13}^2} & & \\ \boxed{\mathbf{k}_{21} \mathbb{W}_{12}^1 \quad \mathbf{k}_{21} \mathbb{W}_{12}^2} & & & \\ & \boxed{\mathbf{k}_{31} \mathbb{W}_{13}^1 \quad \mathbf{k}_{31} \mathbb{W}_{13}^2} & & \end{pmatrix} \begin{array}{l} \text{Sous-domaine 1} \\ \text{Sous-domaine 2} \\ \text{Sous-domaine 3} \end{array} \quad (3.33)$$

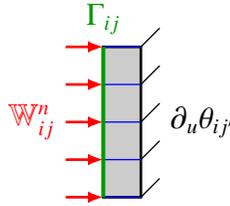
Ainsi pour chaque sous-domaine Ω_i il est effectué en parallèle l'algorithme 2.1.

Algorithme 2.1 : $\mathbf{kA}^T \mathbb{W}$ pour le sous-domaine Ω_i

for $j \in \text{sous-domaines voisins}$ **do**

for $n \in [1 \dots \text{nombre de modes macro}]$ **do**

 Résoudre un problème de Dirichlet sur la semelle ij à déplacement imposé \mathbb{W}_{ij}^n sur l'interface Γ_{ij} :



 Retourner les réactions nodales à l'interface Γ_{ij} .

Le calcul de $\mathbf{N}(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{kA}^T \mathbb{W}$ est effectué de la même façon. Dans ce cas, il n'est plus résolu un problème sur une seule semelle mais sur l'ensemble du sous-domaine Ω_i et de ses semelles. Une condition de Neumann $\mathbf{k}_{ij} \mathbb{W}_{ij}^n$ est imposée sur l'interface Γ_{ij} . Les autres interfaces sont à conditions de Neumann nulles. L'algorithme correspondant est décrit à l'algorithme 2.2 :

Dans ce cas on a l'expression de $\mathbf{N}(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{kA}^T \mathbb{W}$ qui est déterminée de façon parallèle sur les sous-domaines (Algorithme 2.2) :

$$\mathbf{N}(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{kA}^T \mathbb{W} = \begin{pmatrix} \boxed{\mathbf{N}_{12} \mathbb{K}_1^{-1} \mathbf{N}_{12}^T \mathbf{k}_{12} \mathbb{W}_{12}^1 \quad \mathbf{N}_{12} \mathbb{K}_1^{-1} \mathbf{N}_{12}^T \mathbf{k}_{12} \mathbb{W}_{12}^2 \quad \mathbf{N}_{12} \mathbb{K}_1^{-1} \mathbf{N}_{13}^T \mathbf{k}_{13} \mathbb{W}_{13}^1 \quad \mathbf{N}_{12} \mathbb{K}_1^{-1} \mathbf{N}_{13}^T \mathbf{k}_{13} \mathbb{W}_{13}^2} & & \\ \boxed{\mathbf{N}_{13} \mathbb{K}_1^{-1} \mathbf{N}_{12}^T \mathbf{k}_{12} \mathbb{W}_{12}^1 \quad \mathbf{N}_{13} \mathbb{K}_1^{-1} \mathbf{N}_{12}^T \mathbf{k}_{12} \mathbb{W}_{12}^2 \quad \mathbf{N}_{13} \mathbb{K}_1^{-1} \mathbf{N}_{13}^T \mathbf{k}_{13} \mathbb{W}_{13}^1 \quad \mathbf{N}_{13} \mathbb{K}_1^{-1} \mathbf{N}_{13}^T \mathbf{k}_{13} \mathbb{W}_{13}^2} & & \\ \boxed{\mathbf{N}_{21} \mathbb{K}_2^{-1} \mathbf{N}_{21}^T \mathbf{k}_{21} \mathbb{W}_{21}^1 \quad \mathbf{N}_{21} \mathbb{K}_2^{-1} \mathbf{N}_{21}^T \mathbf{k}_{21} \mathbb{W}_{21}^2} & \text{Sous-domaine 2} & \\ \text{Sous-domaine 3} & \boxed{\mathbf{N}_{31} \mathbb{K}_3^{-1} \mathbf{N}_{31}^T \mathbf{k}_{31} \mathbb{W}_{31}^1 \quad \mathbf{N}_{31} \mathbb{K}_3^{-1} \mathbf{N}_{31}^T \mathbf{k}_{31} \mathbb{W}_{31}^2} & \end{pmatrix} \quad (3.34)$$

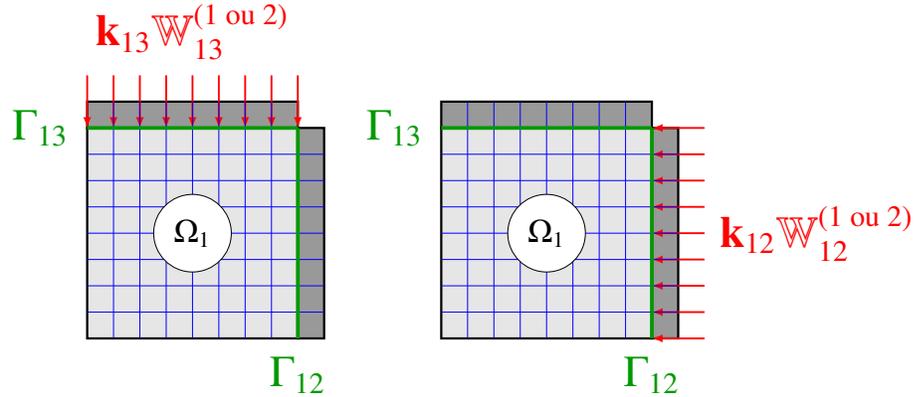
Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages

Algorithme 2.2 : $\mathbf{N}(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{k} \mathbf{A}^T \mathbb{W}$ pour le sous-domaine Ω_i

for $j \in$ sous-domaines voisins **do**

for $n \in [1 \dots \text{nombre de modes macro}]$ **do**

Résoudre un problème de Neumann sur le sous-domaine Ω_i à effort imposé $\mathbf{k}_{ij} \mathbb{W}_{ij}^n$ sur l'interface Γ_{ij} :



Exemple pour le sous-domaine Ω_1 : pour chaque interface (Γ_{12} et Γ_{13}) il est résolu un problème sur $\Omega_1 + \theta_{12} + \theta_{13}$ pour chaque mode macro de l'interface.

Retourner les champs de déplacement aux interfaces Γ_{ij} .

Pour chaque interface Γ_{ik} d'un sous-domaine Ω_i , on extrait les déplacements obtenus par la résolution du problème de Neumann avec une condition imposée sur Γ_{ij} . Par exemple dans l'équation 3.34 les termes (1,1) et (2,1) de la matrice correspondent respectivement aux déplacements sur les interfaces Γ_{12} et Γ_{13} obtenus par la résolution du problème à chargement imposé $\mathbf{k}_{12} \mathbb{W}_{12}^1$ sur Γ_{12} .

Remarque 3.6

Les boucles internes qui consistent à itérer sur les modes macroscopiques par interface peuvent être effectuées par une résolution multi-second membre plus rapide. En effet l'ensemble des degrés de liberté impactés par les conditions de Dirichlet ne change pas d'un mode macro à l'autre.

L'opérateur \mathbf{A} qui intervient ensuite dans l'expression de \mathbf{K}_M va impliquer une communication entre les sous-domaines pour échanger leurs contributions. Pour ce faire, l'opérateur macroscopique est décomposé en contribution par sous-domaine déterminée de façon parallèle :

$$\mathbf{K}_M = \mathbf{K}_M^1 + \mathbf{K}_M^2 + \mathbf{K}_M^3 \quad (3.35)$$

La somme sur tous les sous-domaines est réalisée via la fonction `allreduce` du communicateur MPI.

2.3.3 Bilan sur l'algorithme de résolution

Nous présentons ici un bilan sur l'approche multi-échelle à l'algorithme 2.3

Algorithme 2.3 : Résumé de l'approche multi-échelle pour assurer un équilibre global

Input : Initialisation

Factorisation de l'opérateur macroscopique :

$$\mathbf{K}_M = \mathbb{W}^T \mathbf{A} \left(\mathbf{k}^- \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T - \mathbf{I} \right) \mathbf{k}^- \mathbf{A}^T \mathbb{W}$$

Sauvegarde de l'opérateur intermédiaire :

$$\mathbf{S} = - (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{k}^- \mathbf{A}^T \mathbb{W}$$

while critère d'erreur < objectif **do**

Étape locale :

foreach *Interface* **do**

 └ Résolution du problème d'interface détaillée au chapitre 2

Étape linéaire :

foreach *subdomain* **do**

 1. *Résolution en parallèle sur les sous-domaines*

$$\mathbf{u}_1 = (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \left[\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) \right]$$

 2. *Communication entre les sous-domaines et seconde résolution en parallèle*

$$\text{Communication pour calculer } \psi = \mathbb{W}^T \mathbf{A} \mathbf{k}^- (\mathbf{N} \mathbf{u}_1 - \widehat{\mathbf{W}})$$

$$\text{Sauvegarde du multiplicateur : } \alpha = \mathbf{K}_M^{-1} \psi$$

$$\text{Calcul en parallèle : } \mathbf{u}_2 = \mathbf{S} \alpha$$

$$\text{Finalement : } \mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$$

 3. *Calcul des grandeurs d'interface*

 Direction de recherche pour déterminer les efforts d'interface :

$$\mathbf{F} = \widehat{\mathbf{F}} - \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) - \mathbf{k}^- \mathbf{A}^T \mathbb{W} \alpha$$

 4. *Étape de relaxation*

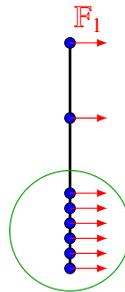
$$s_{n+1} \leftarrow \mu s_{n+1} + (1 - \mu) s_n;$$

3 Approche multi-échelle pour une continuité globale

3.1 Définition d'une base macroscopique en effort

Avec la même idée que la définition de la base macroscopique en déplacement, une base macroscopique en effort est définie pour assurer une continuité globale des champs de déplacement. La base macroscopique associée à l'interface $\Gamma_{EE'}$ est notée $\mathbb{F}_{EE'}$. \mathbb{F} est la base macroscopique en effort concaténée.

Cette base macroscopique comporterait des modes tels que des résultantes et des moments. Ces grandeurs intégrées sur les interfaces ne sont cependant pas aussi triviales à obtenir que les modes macroscopiques en déplacement. En effet pour obtenir une même résultante, plusieurs choix de modes discrétisés sur les interfaces sont possibles. Les modes ne peuvent être indépendants des discrétisations des interfaces. Par exemple des modes moyennés par noeud pour des interfaces non-régulières peuvent conduire à des modes macroscopiques non réguliers (Figure 3.10) et potentiellement perturber la convergence de la méthode.



Concentration artificielle d'effort

FIGURE 3.10 – Exemple de mode macroscopique non régulier

On choisit de définir les modes macroscopiques en effort à partir de ceux en déplacement. Des problèmes de Dirichlet sur les semelles sont résolus en utilisant les modes macroscopiques en déplacement comme conditions limites :

$$\mathbb{F} = (\mathbf{A}\mathbf{k}^{-1}\mathbf{A}^T) \mathbb{W} \quad (3.36)$$

3.2 Définition de l'étape linéaire

3.2.1 Ecriture du problème

De même que pour l'approche en équilibre, la continuité globale est imposée par l'ajout d'une nouvelle relation :

$$\mathbb{F}^T \mathbf{B}\mathbf{W} = 0 \quad (3.37)$$

La relation de direction de recherche est écrite de façon globale comme un problème de minimisation sous contrainte.

$$\min_{\mathbb{F}^T \mathbf{B}\mathbf{W}=0} \left\{ (\mathbf{W} - \widehat{\mathbf{W}})^T (\mathbf{F} - \widehat{\mathbf{F}}) + \frac{1}{2} (\mathbf{W} - \widehat{\mathbf{W}})^T \mathbf{k}^{-1} (\mathbf{W} - \widehat{\mathbf{W}}) \right\} \quad (3.38)$$

On définit le lagrangien $\mathcal{L}(\mathbf{W}, \beta)$ dont β est le multiplicateur associé à la contrainte $\mathbb{F}^T \mathbf{B} \mathbf{W} = 0$.

$$\mathcal{L}(\mathbf{W}, \beta) = (\mathbf{W} - \widehat{\mathbf{W}})^T (\mathbf{F} - \widehat{\mathbf{F}}) + \frac{1}{2} (\mathbf{W} - \widehat{\mathbf{W}})^T \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) + \beta^T \mathbb{F}^T \mathbf{B} \mathbf{W} \quad (3.39)$$

Cela conduit à la relation de direction de recherche globale :

$$\mathbf{F} - \widehat{\mathbf{F}} + \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) + \mathbf{B}^T \mathbb{F} \beta = 0 \quad (3.40)$$

Ainsi on obtient le problème à résoudre à l'étape linéaire :

Trouver (\mathbf{W}, \mathbf{F}) solution de :

$$\begin{cases} \mathbf{K} \mathbf{u} = \mathbf{f}_d + \mathbf{N}^T \mathbf{F} \\ \mathbf{W} = \mathbf{N} \mathbf{u} \\ \mathbb{F}^T \mathbf{B} \mathbf{W} = 0 \\ (\mathbf{F} - \widehat{\mathbf{F}}) + \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) + \mathbf{B}^T \mathbb{F} \beta = 0 \end{cases} \quad (3.41)$$

3.2.2 Obtention du problème en déplacement

L'expression du multiplicateur en fonction de l'inconnue en effort \mathbf{F} est obtenue via la relation de continuité globale $\mathbb{F}^T \mathbf{B} \mathbf{W} = 0$:

$$\beta = -(\mathbb{F}^T \mathbf{B} \mathbf{k}^{-1} \mathbf{B}^T \mathbb{F})^{-1} [\mathbf{k}^{-1} (\mathbf{F} - \widehat{\mathbf{F}}) - \widehat{\mathbf{W}}] \quad (3.42)$$

Injecter cette expression dans la relation de direction de recherche permet d'obtenir une relation entre le champ de déplacement \mathbf{W} et le champ d'effort \mathbf{F} :

$$(\mathbf{I} - \mathbf{Q}_M) [\mathbf{F} - \widehat{\mathbf{F}} - \mathbf{k}^- \widehat{\mathbf{W}}] + \mathbf{k}^- \mathbf{W} = 0 \quad (3.43)$$

avec $\mathbf{Q}_M = \mathbf{B}^T \mathbb{F} (\mathbb{F}^T \mathbf{B} \mathbf{k}^{-1} \mathbf{B}^T \mathbb{F})^{-1} \mathbb{F}^T \mathbf{B} \mathbf{k}^{-1}$ un projecteur macroscopique qui, au même titre que \mathbf{P}_M , ne dépend que des directions de recherche. La direction de recherche écrite ainsi ne permet pas d'obtenir de relation directe entre \mathbf{F} et \mathbf{W} . En effet seule la partie microscopique $(\mathbf{I} - \mathbf{Q}_M) \mathbf{F}$ est liée au déplacement. De plus suivant l'expression de \mathbf{Q}_M , la partie macroscopique des efforts $\mathbf{Q}_M \mathbf{F}$ peut être écrite sous la forme :

$$\mathbf{Q}_M \mathbf{F} = \mathbf{B}^T \mathbb{F} \gamma \quad (3.44)$$

Cela permet d'introduire cette direction de recherche au sein de la relation d'équilibre des sous-domaines :

$$\mathbf{K} \mathbf{u} = \mathbf{f}_d + \underbrace{\mathbf{N}^T (\mathbf{I} - \mathbf{Q}_M) \mathbf{F}}_{\text{partie micro}} + \underbrace{\mathbf{N}^T \mathbf{Q}_M \mathbf{F}}_{\text{partie macro : } \mathbf{B}^T \mathbb{F} \gamma} \quad (3.45)$$

Nous obtenons donc un problème à deux inconnues γ et \mathbf{u} à résoudre.

⚠ Remarque 3.7

D'après l'expression de \mathbf{Q}_M , et puisque $\mathbf{B}\widehat{\mathbf{W}} = 0$, alors $\mathbf{Q}_M \mathbf{k}^- \widehat{\mathbf{W}} = 0$

γ est déterminé en utilisant la relation de continuité globale après avoir exprimé \mathbf{W} en fonction γ .

$$\gamma = - \left(\mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{B}^T \mathbb{F} \right)^{-1} \mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \left[\mathbf{f}_d + \mathbf{N}^T (\mathbf{I} - \mathbf{Q}_M) \widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}} \right] \quad (3.46)$$

En réinjectant cette expression de γ , le problème en déplacement devient :

$$\begin{aligned} & \left(\mathbf{K} + \tilde{\mathbf{k}} \right) \mathbf{u} = (\mathbf{I} - \mathbf{R}) \left[\mathbf{f}_d + \mathbf{N}^T (\mathbf{I} - \mathbf{Q}_M) \widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}} \right] \\ & \text{avec } \mathbf{R} = \mathbf{N}^T \mathbf{B}^T \mathbb{F} \left(\mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{B}^T \mathbb{F} \right)^{-1} \mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \end{aligned} \quad (3.47)$$

Le projecteur est ici appliqué aux efforts d'interface : $(\mathbf{I} - \mathbf{Q}_M) \widehat{\mathbf{F}}$, contrairement à l'approche en équilibre où il est appliqué aux déplacements : $(\mathbf{I} - \mathbf{P}_M) \widehat{\mathbf{W}}$. De plus, dans cette approche, le second membre est corrigé par un opérateur \mathbf{R} alors que l'opérateur de rigidité n'est pas impacté. Il devient donc plus délicat d'interpréter cette approche en continuité globale par une modification de la direction de recherche mono-échelle.

3.3 Schéma de résolution du problème en déplacement

Un schéma à deux échelles est mis en place pour résoudre le problème en déplacement de l'équation 3.47. Pour visualiser cela, l'équation 3.47 peut se ré-écrire sous la forme :

$$\mathbf{u} = \underbrace{(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \left[\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) \right]}_{\text{solution mono-échelle}} - \underbrace{(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{B}^T \mathbb{F} \mathbf{K}_M^{-1} \mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \left[\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}}) \right]}_{\text{solution mono-échelle}} \quad (3.48)$$

avec \mathbf{K}_M l'opérateur macroscopique :

$$\mathbf{K}_M = \mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{B}^T \mathbb{F} \quad (3.49)$$

Un algorithme équivalent à celui proposé dans le cadre de l'approche multi-échelle en équilibre global est mis en place. On résout tout d'abord un problème mono-échelle dont la solution est utilisée pour résoudre un problème macroscopique. Cette approche est résumée à l'algorithme 3.1.

Algorithme 3.1 : Résumé de l'approche multi-échelle pour assurer un équilibre global

Input : Initialisation

Factorisation de l'opérateur macroscopique :

$$\mathbf{K}_M = \mathbb{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{B}^T \mathbb{F}$$

Sauvegarde de l'opérateur intermédiaire :

$$\mathbf{S} = -(\mathbf{K} + \tilde{\mathbf{k}})^{-1} \mathbf{N}^T \mathbf{B}^T \mathbb{F}$$

while critère d'erreur < objectif **do**

Étape locale :

foreach Interface **do**

 └ Résolution du problème d'interface détaillée au chapitre 2

Étape linéaire :

foreach Sous-domaine **do**

 1. *Résolution en parallèle sur les sous-domaines*

$$\mathbf{u}_1 = (\mathbf{K} + \tilde{\mathbf{k}})^{-1} [\mathbf{f}_d + \mathbf{N}^T (\widehat{\mathbf{F}} + \mathbf{k}^- \widehat{\mathbf{W}})]$$

 2. *Communication entre les sous-domaines et seconde résolution en parallèle*

$$\text{Communication pour calculer } \psi = \mathbb{F}^T \mathbf{B} (\mathbf{N} \mathbf{u}_1 - \widehat{\mathbf{W}})$$

$$\text{Sauvegarde du multiplicateur : } \beta = \mathbf{K}_M^{-1} \psi$$

$$\text{Calcul en parallèle : } \mathbf{u}_2 = \mathbf{S} \beta$$

$$\text{Finalement : } \mathbf{u} = \mathbf{u}_1 + \mathbf{u}_2$$

 3. *Calcul des grandeurs d'interface*

 Direction de recherche pour déterminer les efforts d'interface :

$$\mathbf{F} = \widehat{\mathbf{F}} - \mathbf{k}^- (\mathbf{W} - \widehat{\mathbf{W}}) - \mathbf{B}^T \mathbb{F} \beta$$

 4. *Étape de relaxation*

$$s_{n+1} \leftarrow \mu s_{n+1} + (1 - \mu) s_n;$$

4 Comparaison des deux approches - exemples numériques

4.1 Étude de l'extensibilité

Nous étudions l'effet de l'approche multi-échelle sur un cas académique 3D (Figure 3.11). on considère une structure élancée encastrée à une extrémité et soumise à un chargement de traction à l'extrémité opposée. Elle est décomposée en 8, 16, 32, 64 et 128 sous-domaines. La direction de recherche est la même pour toutes les interfaces. Les différents paramètres sont précisés dans le tableau 3.1.

La figure 3.13 montre la comparaison entre l'implémentation mono-échelle et les implémentations multi-échelles pour différentes sous-structurations. On observe dans le cas mono-échelle une dépendance à la décomposition (Figure 3.13a) : plus le nombre de sous-structures augmente, plus il est nécessaire d'itérer pour faire propager l'information des conditions limites à l'ensemble de la structure (Figure 3.13a).

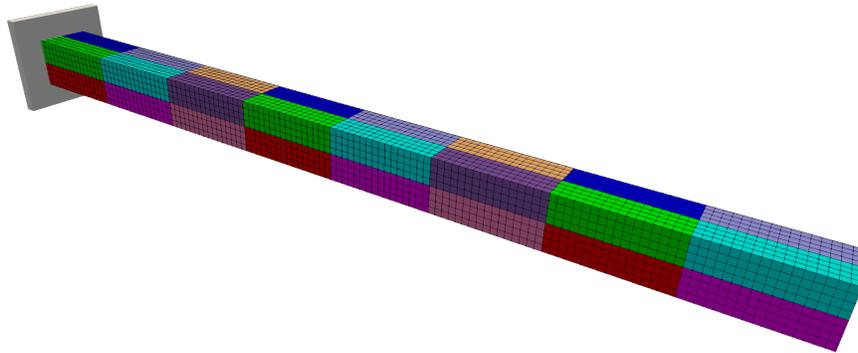


FIGURE 3.11 – Poutre 3D

Paramètres	Valeur
Module de Young E	200 GPa
Module de Young des semelles E_S	0.0078125 E
ν	0.3
Longueur L	160 mm
Section S	10×10 mm
Chargement F_d	10 MPa
Taille de maille h	1.25 mm
Sous-structuration	8,16,32,64,128
Nombre de noeuds	23825

TABLE 3.1 – Paramètres

L'approche multi-échelle en équilibre permet de faire propager les informations globales des conditions limites à travers l'ensemble de la structure. Cela permet de gagner en extensibilité. La figure 3.13b l'illustre très bien : quelque soit la sous-structuration, la convergence est sensiblement identique. Une forte accélération sur les premières itérations traduit la transmission des informations globales à l'ensemble de la structure. L'ensemble des sous-domaines est impacté. A partir de quelques itérations cette transmission globale n'a plus d'influence. Seulement des effets de communication locale entre sous-domaines impactent la convergence. A la figure 3.12 nous présentons les premières itérations de l'approche multi-échelle en équilibre. Nous vérifions l'équilibre global à travers la propagation immédiate des contraintes à travers la structure. Les sous-domaines se déforment suivant les informations macroscopiques des conditions limites et permettent de converger à la 30^e itération vers un déplacement maximal en bout de poutre de 0.00802mm correspondant au calcul de référence.

L'approche en continuité n'est pas aussi efficace que l'approche en équilibre. L'extensibilité n'est pas obtenue et une dépendance à la sous-structuration est observée (Figure 3.13c). Il faut cependant noter une très légère amélioration par rapport à l'approche mono-échelle due à la vérification de la condition de continuité globale.

Dans le cadre de l'approche en équilibre, l'extensibilité est acquise grâce à la perturbation de

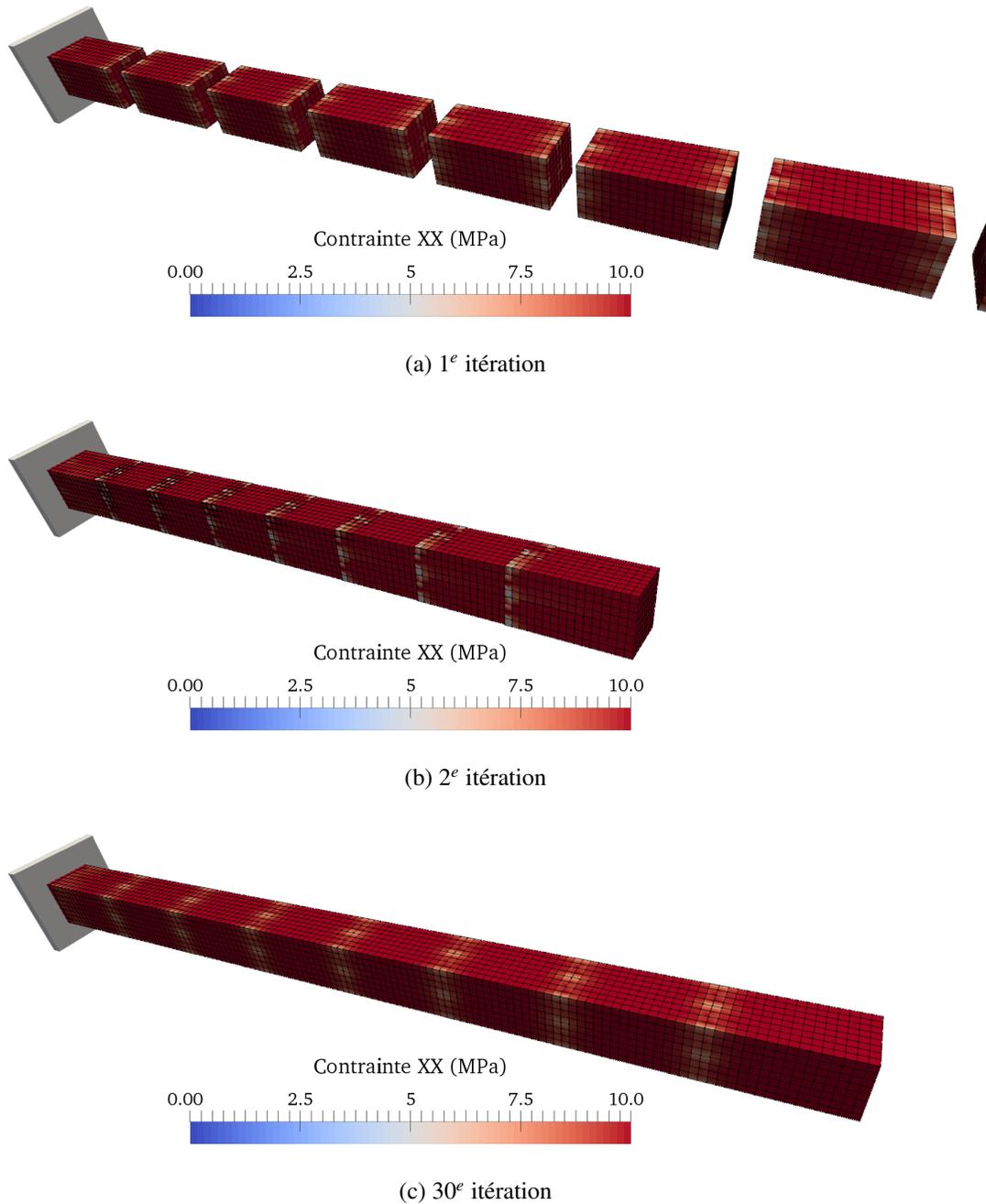


FIGURE 3.12 – Déformée ($\times 500$) et contrainte σ_{xx} pour l’approche en équilibre à 32 sous-domaines

l’opérateur de rigidité $(\mathbf{K} + \tilde{\mathbf{k}})$ par un projecteur macroscopique \mathbf{P}_M . Ce projecteur venant coupler les sous-domaines voisins, l’ensemble des sous-domaines se retrouvent "liés" et l’inversion du problème fait propager l’information des conditions limites à l’ensemble des sous-structures.

Cependant dans le cas de l’approche en continuité globale l’opérateur de rigidité n’est pas impacté par le projecteur macroscopique \mathbf{Q}_M et les sous-domaines restent indépendants les uns des

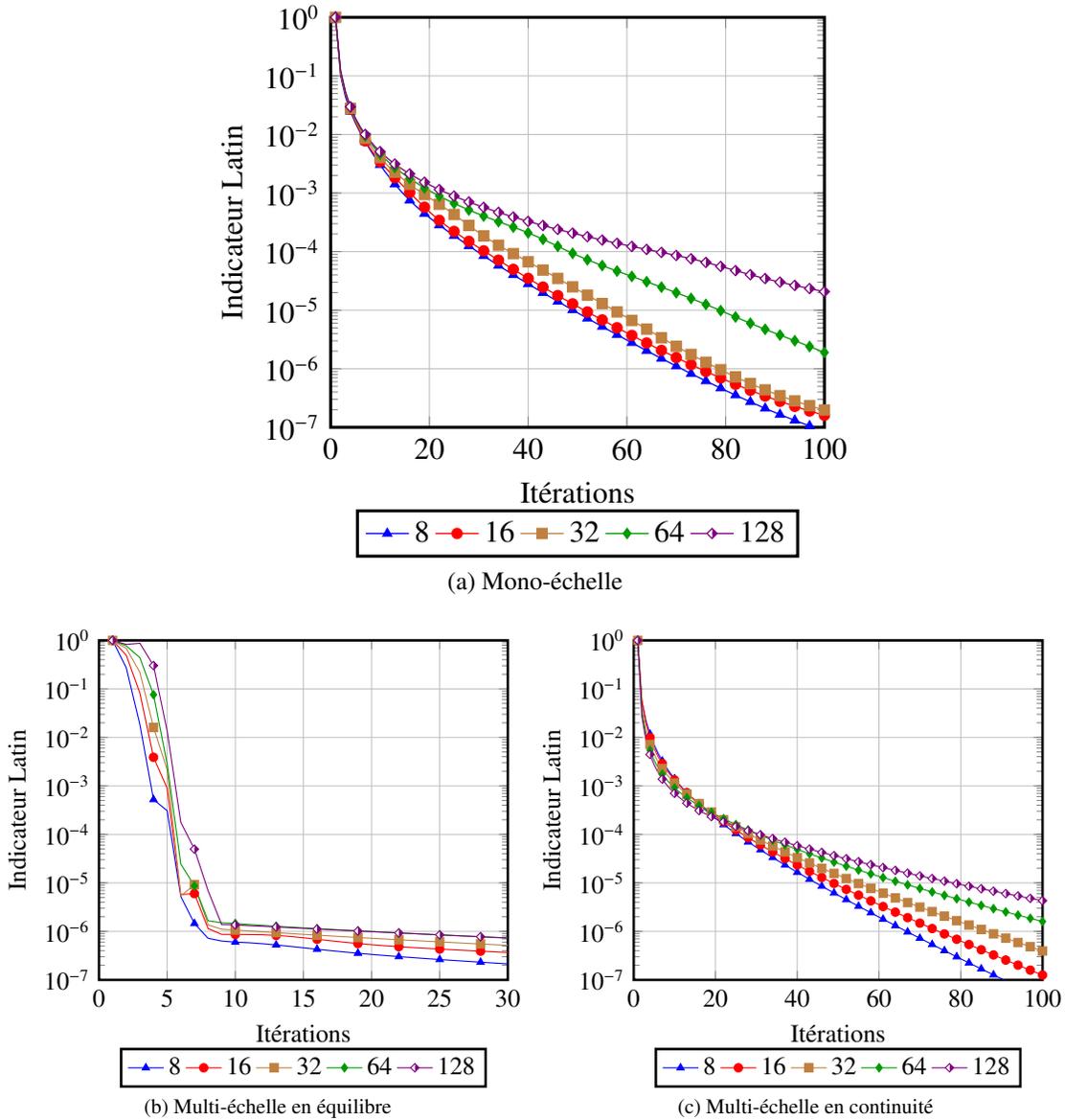


FIGURE 3.13 – Comparaison entre l'approche mono-échelle et multi-échelle

autres. Un couplage intervient cependant via le second membre du problème. L'opérateur \mathbf{R} fait intervenir l'opérateur macroscopique \mathbf{K}_M .

$$\mathbf{R} = \mathbf{N}^T \mathbf{B}^T \mathbf{F} \mathbf{K}_M^{-1} \mathbf{F}^T \mathbf{B} \mathbf{N} (\mathbf{K} + \tilde{\mathbf{k}})^{-1} \quad (3.50)$$

Cela explique une légère transmission des conditions limites à l'ensemble des sous-domaines. Cependant le couplage qui n'apparaît pas dans l'opérateur de rigidité pénalise fortement la convergence. Les premières itérations sont présentées à la figure 3.14. Il est bien vérifié une continuité globale des sous-domaines. Cependant nous observons que la contrainte ne se propage que lentement à l'ensemble de la structure. Par exemple il est présenté la carte de contrainte à la 30^e itération

et il est observé qu'elle n'est pas encore homogène sur l'ensemble de la structure. De plus le déplacement maximal en bout de poutre est de 0.00573mm, encore éloigné de la solution de référence.

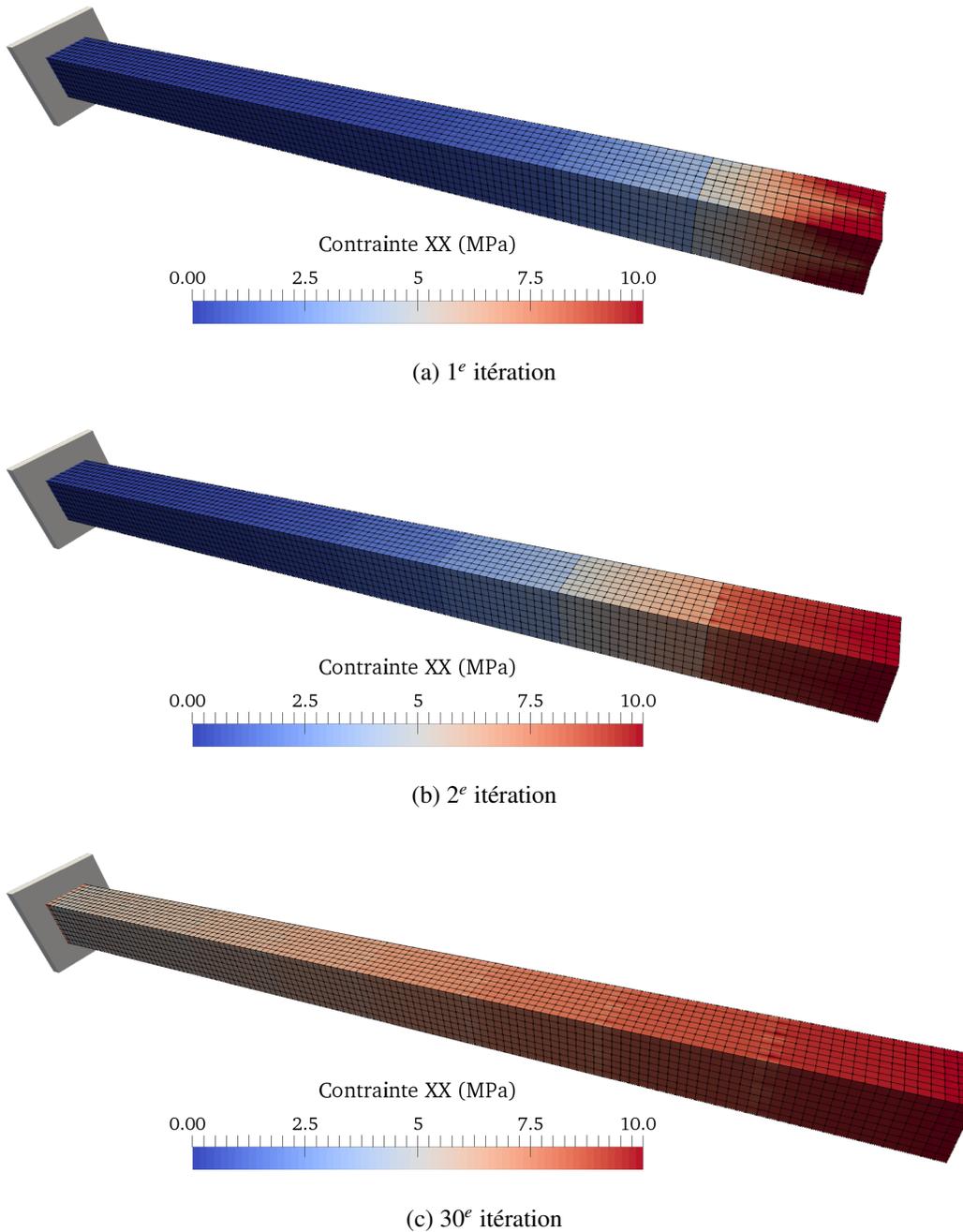


FIGURE 3.14 – Déformée ($\times 2000$) et contrainte σ_{xx} pour l'approche en continuité

4.2 Influence de la direction de recherche

Nous étudions maintenant l'influence de la direction de recherche sur l'approche multi-échelle. Nous considérons pour cela la structure décomposée en 32 sous-domaines. Nous pondérons le module de Young des semelles précédemment choisi par 0.01, 0.1, 10 et 100. Nous présentons les différentes convergences en terme de saut de déplacement et de déséquilibre d'effort (Figure 3.15).

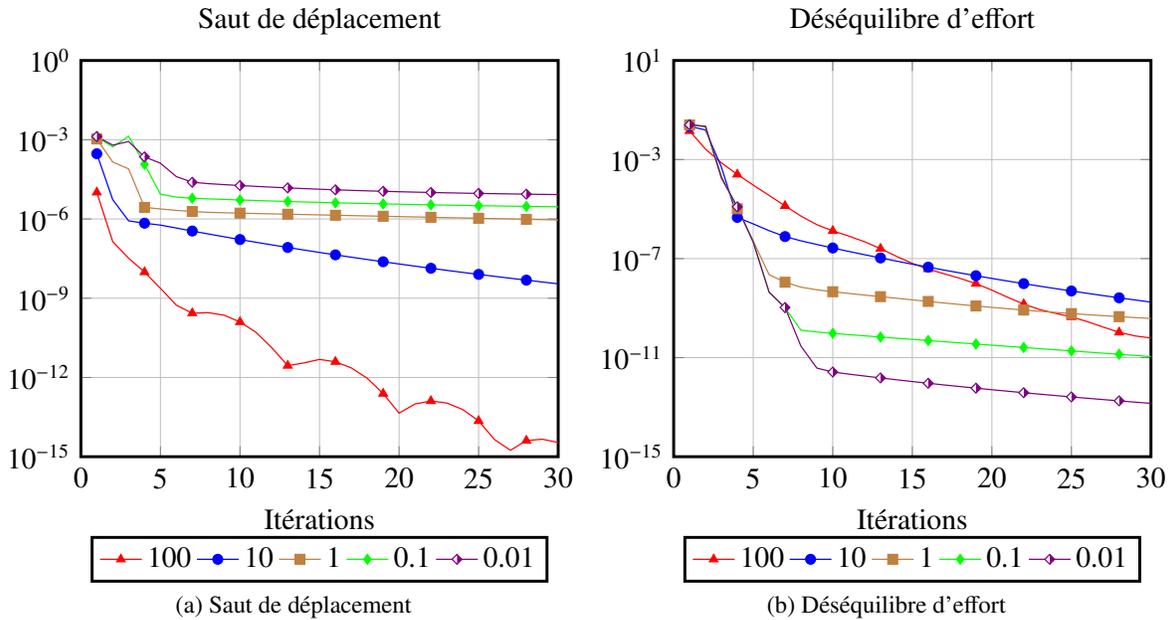


FIGURE 3.15 – Approche en équilibre

Ces résultats peuvent s'interpréter facilement en considérant les cas limites de la relation de direction de recherche :

- Cas où $\mathbf{k}^- \rightarrow \infty$: Avec une telle hypothèse la direction de recherche est équivalente à :

$$\mathbf{W} - \widehat{\mathbf{W}} + \mathbf{A}^T \mathbb{W} \alpha = 0 \quad (3.51)$$

Une telle direction de recherche améliore la convergence en déplacement. Les efforts deviennent négligeables et les déplacements à l'étape linéaire tendent vers les déplacements déterminés lors de l'étape locale. Comme les efforts sont négligés, l'indicateur sur le déséquilibre des sous-domaines est détérioré. Cependant avec l'approche en équilibre qui aide à vérifier globalement l'équilibre des interfaces, cela reste raisonnable.

- Cas où $\mathbf{k}^- \rightarrow 0$: la direction de recherche devient équivalente à :

$$\mathbf{F} - \widehat{\mathbf{F}} = 0 \quad (3.52)$$

Les efforts de l'étape linéaire sont égaux à ceux obtenus lors de l'étape locale. Ils sont par conséquent quasi-équilibrés et ainsi l'indicateur de déséquilibre diminue (Figure 3.15b). Bien entendu cela se fait au détriment de la convergence en déplacement.

Bien que l’extensibilité n’ait pu être vérifiée dans le cadre de l’approche en continuité globale, nous effectuons aussi une étude de l’influence de la direction de recherche sur la convergence (Figure 3.16). Les analyses précédentes sur les cas limites de la relation de direction de recherche restent valables. Une augmentation de la pondération de la direction de recherche donne plus de poids aux grandeurs de déplacement et détériore la convergence en effort. Cette détérioration n’est pas rattrapée par un problème macroscopique contrairement à l’approche précédente. Une diminution de la pondération se traduit par une nette amélioration de la convergence en effort (Figure 3.16b). En revanche le saut de déplacement est dégradé. Seul le problème macroscopique permet d’imposer une certaine continuité à la structure.

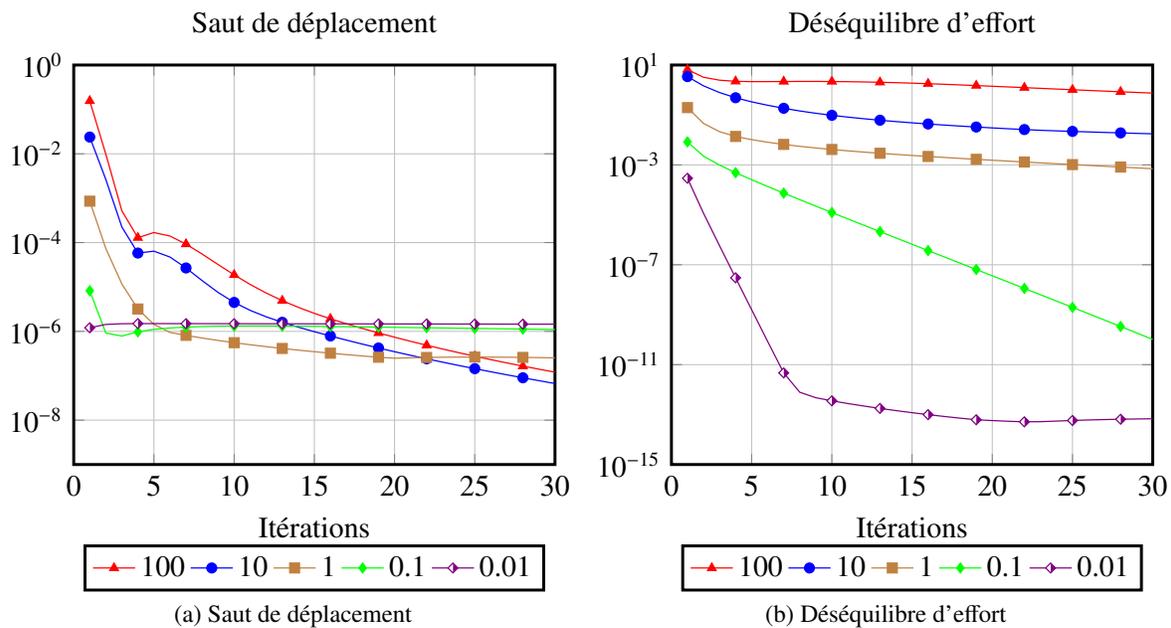


FIGURE 3.16 – Approche en continuité

Nous présentons à la figure 3.17 les indicateurs d’erreur Latin pour chaque pondération des deux approches multi-échelles. Avec cette comparaison sur les indicateurs d’erreur Latin, nous retrouvons les conclusions de [Loiseau, 2001] qui affirmait une optimalité de l’approche en équilibre pour de grandes directions de recherche alors que l’approche en continuité s’avère plus performante pour les faibles directions de recherche. Cependant les continuités des grandeurs d’interface n’ont pas été étudiées. Des indicateurs Latin à même niveau d’erreur peuvent représenter des cas de convergence bien différents. Une étude de l’extensibilité de l’approche en continuité n’y était pas non plus proposée.

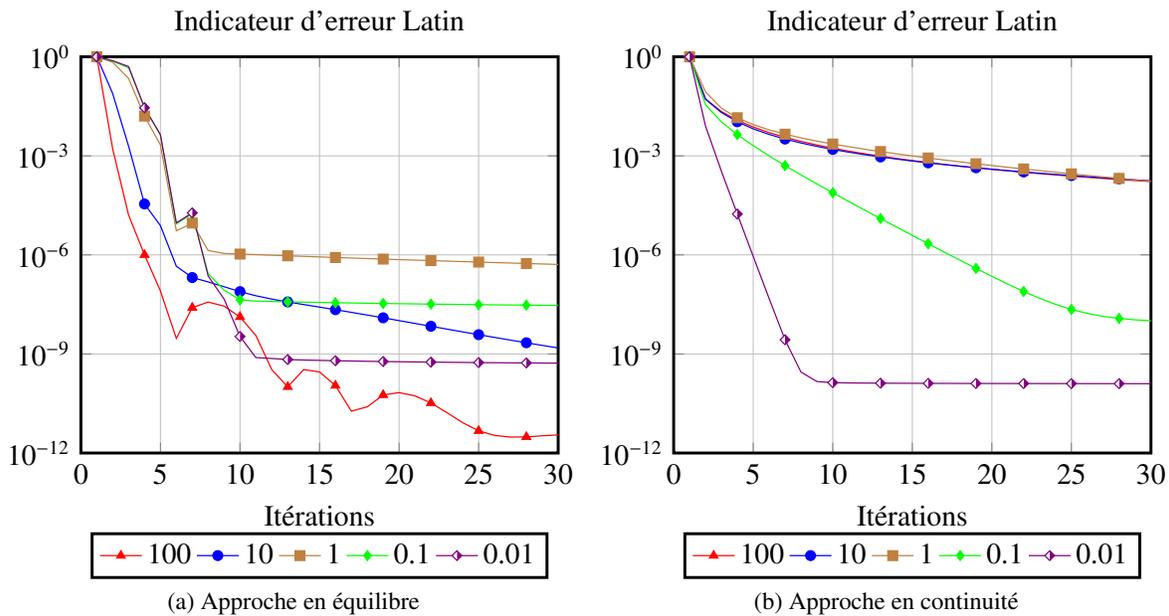


FIGURE 3.17 – Approche en continuité

4.3 Influence du paramètre de relaxation

Nous nous intéressons ici au paramètre de relaxation. Il est démontré que la méthode Latin en décomposition de domaine pour une approche mono-échelle converge pour un paramètre de 0.5. Il est pourtant admis dans la communauté qu'un choix de 0.8 permet d'améliorer la convergence. Nous illustrons ici le fait que cette affirmation n'est pas nécessairement vérifiée pour l'approche multi-échelle. Pour illustrer cela nous restons dans le cas d'application de la poutre sous-structurée en 32 sous-domaines.

Nous observons à la figure 3.18a une dépendance claire au paramètre de relaxation. Plus celui-ci est faible plus la convergence est ralentie mais régulière. Dans le cas où ce paramètre est augmenté, des difficultés de convergence apparaissent et parfois des cas de divergence se manifestent. Le choix de 0.5 pour lequel la convergence a été démontrée est celui qui doit être choisi en multi-échelle.

Le choix empirique de 0.8 est fondé sur des approches mono-échelles. Par exemple à la figure 3.18b nous représentons l'influence de la relaxation pour l'approche mono-échelle. Il semble effectivement qu'une relaxation plus élevée que 0.5 permet une convergence un peu plus rapide.

Au vu des résultats précédents, il paraît difficile de vouloir maintenir un choix de relaxation de 0.8. Il est préférable de chercher à respecter au mieux les conditions de convergence de la méthode Latin mono-échelle.

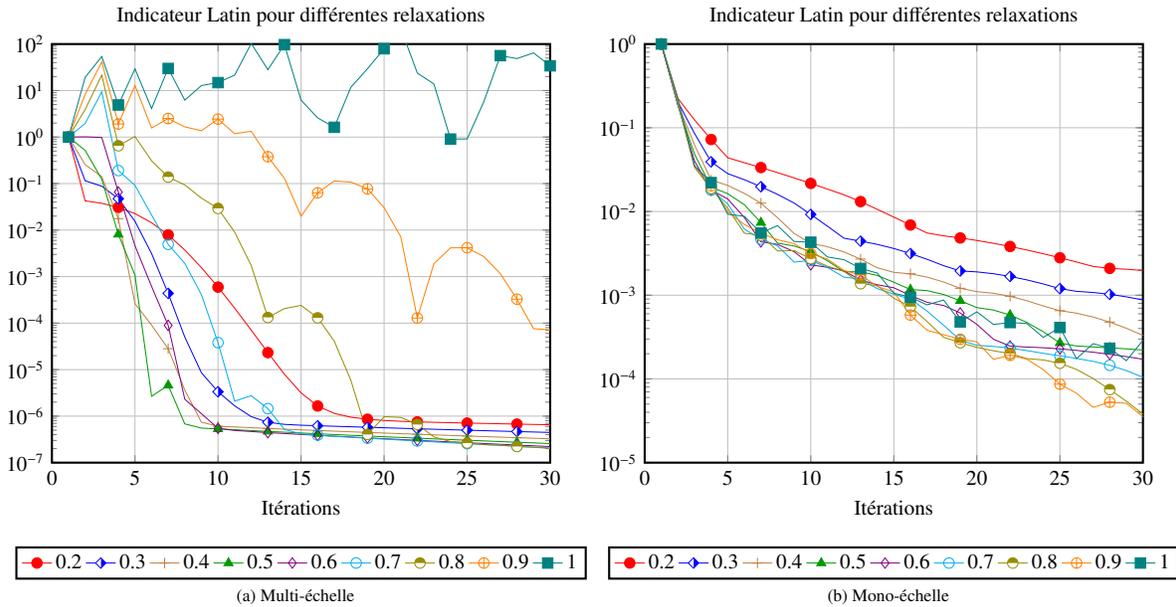


FIGURE 3.18 – Influence du paramètre de relaxation

5 Accélération de type Krylov

5.1 Principe de l'accélération de Krylov

5.1.1 Quelques rappels sur les méthodes de Krylov

Les méthodes de Krylov sont des méthodes de projection pour résoudre un problème linéaire du type $Hx = b$ en cherchant itérativement des solutions x_m sur l'espace affine $x_0 + \mathcal{K}_m$ de dimension m sous la condition : $b - Hx_m \perp \mathcal{L}_m$. Avec \mathcal{L}_m un sous-espace de dimension m .

Les méthodes de Krylov sont caractérisées par le sous-espace dit de Krylov \mathcal{K}_m :

$$\mathcal{K}_m = \text{Vect}(r_0, Hr_0, \dots, H^{m-1}r_0) \quad (3.53)$$

Le choix du sous-espace de contrainte \mathcal{L}_m permet de définir différentes méthodes. Par exemple choisir $\mathcal{L}_m = \mathcal{K}_m$ conduit à la méthode de *Full Orthogonalization Method* et aux nombreux algorithmes qui en découlent comme le *gradient conjugué* pour des opérateurs symétriques, *ORTHOMIN* pour des applications non-symétriques. Le choix $\mathcal{L}_m = H\mathcal{K}_m$ amène aux méthodes type *GMRES*. Une revue générale des solveurs itératifs pour des problèmes linéaires peut être trouvée dans [Saad, 2003].

5.1.2 Application à la méthode Latin

Dans le cas d'une décomposition de domaine avec des interfaces parfaites, l'opérateur des itérés \mathcal{H} qui permet de passer d'une solution S_i à une solution S_{i+1} est constant. L'algorithme de point

fixe de la méthode Latin s'écrit (Figure 3.19) :

$$\text{Trouver } S \text{ telle que : } \mathcal{H}(S) = S \quad (3.54)$$

Cet algorithme de point fixe converge lentement. L'étape de relaxation ne permet pas de prendre en compte l'ensemble des solutions à chaque itération. Nous nous intéressons dans cette section à une accélération de type Krylov qui va permettre de venir optimiser la solution d'une itération par rapport à l'ensemble de celles obtenues précédemment. Le caractère constant de l'opérateur des itérés permet d'insérer assez aisément les itérations Latin dans un solveur de Krylov.

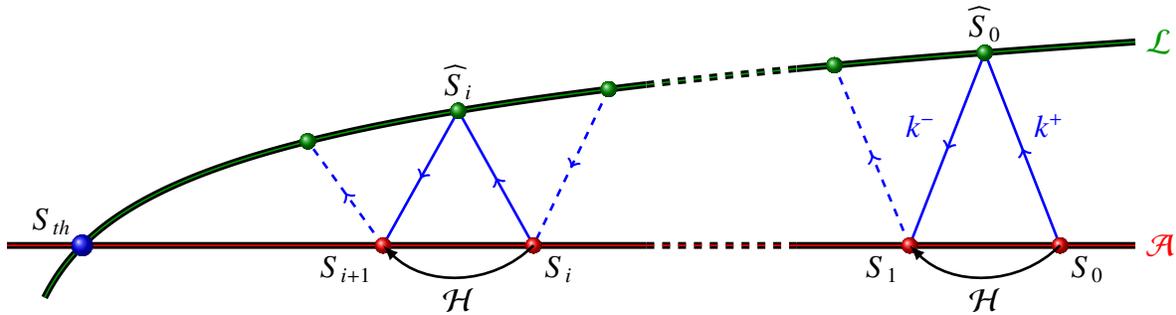


FIGURE 3.19 – Principe de la méthode Latin

Les conditions limites étant imposées directement sur les sous-domaines (section 3.1.5), ainsi l'opérateur des itérés \mathcal{H} s'écrit de manière affine sous la forme :

$$\mathcal{H}(S) = \tilde{H} + \tilde{S}_0 \quad (3.55)$$

avec \tilde{H} un opérateur linéaire et \tilde{S}_0 découlant des conditions limites imposées.

⚠ Remarque 3.8

\tilde{S}_0 n'est autre que la solution obtenue à la première itération sans relaxation ni accélération de Krylov. En effet les grandeurs d'interface sont initialisées à 0 : $S_0 = 0$. Cela implique qu'à la première itération la solution obtenue S_1 n'est autre que $\tilde{H}(0) + \tilde{S}_0 = \tilde{S}_0$. Ainsi il est nécessaire d'effectuer une première itération avant de pouvoir utiliser l'accélération de Krylov.

Le problème à résoudre devient :

$$(\tilde{H} - \mathbb{I})S = -\tilde{S}_0 \quad (3.56)$$

Nous notons finalement le problème pour lequel est mis en place un solveur de Krylov :

$$HS = -\tilde{S}_0 \quad (3.57)$$

Dans notre cas l'opérateur H n'est pas symétrique. Un algorithme de gradient conjugué n'est donc pas adapté. Nous choisissons alors d'utiliser un algorithme de type *ORTHOMIN* : algorithme 5.1.

Algorithme 5.1 : ORTHOMIN**Input :**

$$S_0 = 0$$

$$R_0 = -\mathcal{H}(X_0) = -\tilde{S}_0 \text{ 1}^{\text{e}} \text{ itération Latin}$$

$$P_0 = R_0$$

$$Q_0 = HP_0 = \mathcal{H}(P_0) - P_0 - \tilde{S}_0 \quad 2^{\text{e}} \text{ itération Latin}$$

$$\alpha_0 = \frac{Q_0^T R_0}{Q_0^T Q_0}$$

$$S_1 = S_0 + \alpha_0 P_0 \text{ et } R_1 = R_0 - \alpha_0 Q_0$$

for $i = 1 \dots n$ **do**

$$V = HR_i = \mathcal{H}(R_i) - R_i - \tilde{S}_0 \quad i^{\text{e}} \text{ itération Latin}$$

foreach $j < i$ **do**

$$\beta_j = \frac{Q_j^T V}{Q_j^T Q_j}$$

$$P_i = R_i - \sum_{j=0}^{i-1} \beta_j P_j$$

$$Q_i = V_i - \sum_{j=0}^{i-1} \beta_j Q_j$$

$$\alpha_i = \frac{Q_i^T R_i}{Q_i^T Q_i}$$

$$S_{i+1} = S_i + \alpha_i P_i \text{ et } R_{i+1} = R_i - \alpha_i Q_i$$

Puisque les itérations portent sur le résidu, il n'est pas possible d'avoir accès aux champs de déplacements et d'efforts de l'étape locale. Par conséquent, l'indicateur d'erreur Latin ne peut être calculé. Pour comparer les deux approches, l'accélération de Krylov et la méthode classique, on ré-écrit différemment l'étape de relaxation classique afin de faire apparaître un résidu.

$$\begin{aligned} S_{i+1} &= \mu \tilde{S}_{i+1} + (1 - \mu) S_i \\ &= S_i + \underbrace{\mu (\tilde{S}_{i+1} - S_i)}_{\text{Résidu } R_{i+1}} \end{aligned} \quad (3.58)$$

Les différentes grandeurs de l'algorithme 5.1 regroupent des quantités relatives aux déplacement et en effort. Ainsi il est possible d'écrire par exemple :

$$S_i = \begin{bmatrix} kW_i \\ F_i \end{bmatrix} \text{ ou encore } R_i = \begin{bmatrix} kR_i^W \\ R_i^F \end{bmatrix} \quad (3.59)$$

avec R_i^W et R_i^F respectivement les quantités de déplacement et d'effort du résidu. Les quantités en déplacement sont pondérées par la direction de recherche k afin de garantir l'homogénéité des différentes grandeurs.

⚠ Remarque 3.9

Effectuer une itération Latin permet d'obtenir la grandeur $\mathcal{H}(R_i)$. Pour obtenir $H(R_i)$, il suffit de lui soustraire R_i ainsi que la toute première solution \tilde{S}_0 .

Chaque calcul de pas de descente β_i ou de direction de recherche α_i nécessite une communication globale pour échanger les contributions de chaque sous-domaine. Bien que n'échangeant que des scalaires, la présence d'un très grand nombre de sous-domaines peut ralentir l'algorithme. De plus il est nécessaire de sauvegarder des grandeurs pour chaque itération. En effet il est effectué une ré-orthonormalisation des directions de recherche et des directions conjuguées par rapport à celles déterminées aux itérations précédentes.

5.2 Illustration

Nous illustrons l'implémentation de cette accélération de Krylov sur un premier exemple 2D très simple. Une seconde illustration sur un cas 3D plus complexe est aussi montrée.

5.2.1 Cas 2D

Nous nous plaçons dans un cas de poutre 2D en traction (Figure 3.20). La structure est décomposée en quatre sous-domaines chacun discrétisé en quatre éléments. Seule une approche mono-échelle est utilisée pour cette exemple.

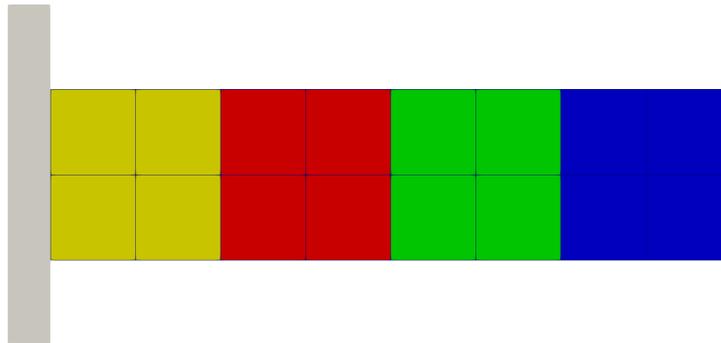


FIGURE 3.20 – Poutre 2D

Dans le cas où le coefficient de Poisson est nul, le problème est équivalent à un problème 1D poutre (Figure 3.21) pour lequel le problème d'interface est ramené à 12 inconnues. On s'attend donc à une convergence exacte en maximum 12 itérations.

A la figure 3.22 il est présenté l'évolution de la norme du résidu R_i au cours des itérations pour différents cas d'étude :

- une accélération de Krylov avec un coefficient de Poisson $\nu = 0$,
- une accélération de Krylov avec un coefficient de Poisson $\nu = 0.3$,

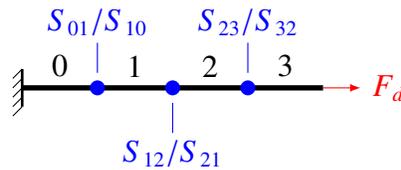


FIGURE 3.21 – Problème équivalent 1D

- une relaxation classique de 0.5 avec un coefficient de Poisson $\nu = 0$,
- une relaxation classique de 0.5 avec un coefficient de Poisson $\nu = 0.3$.

Il est constaté une très nette accélération de la convergence avec l’utilisation d’un solveur de Krylov. Avec l’accélération de Krylov le calcul converge en 6 itérations pour le cas $\nu = 0$ ($\|R_i\| \sim 10^{-30}$), en 18 itérations pour $\nu = 0.3$ ($\|R_i\| \sim 10^{-27}$) alors que même avec 60 itérations l’approche classique avec relaxation atteint péniblement un résidu de 10^{-5} . Il est normal de constater qu’avec une accélération de Krylov le cas pour $\nu = 0$ converge bien plus vite que le cas $\nu \neq 0$. En effet, ce cas avec coefficient de Poisson nul est équivalent à un problème 1D. Ainsi pour chaque interface, seules quatre inconnues sont à considérer alors que pour un coefficient de Poisson non nul 24 inconnues interviennent. Le problème étant plus petit, il converge plus rapidement.

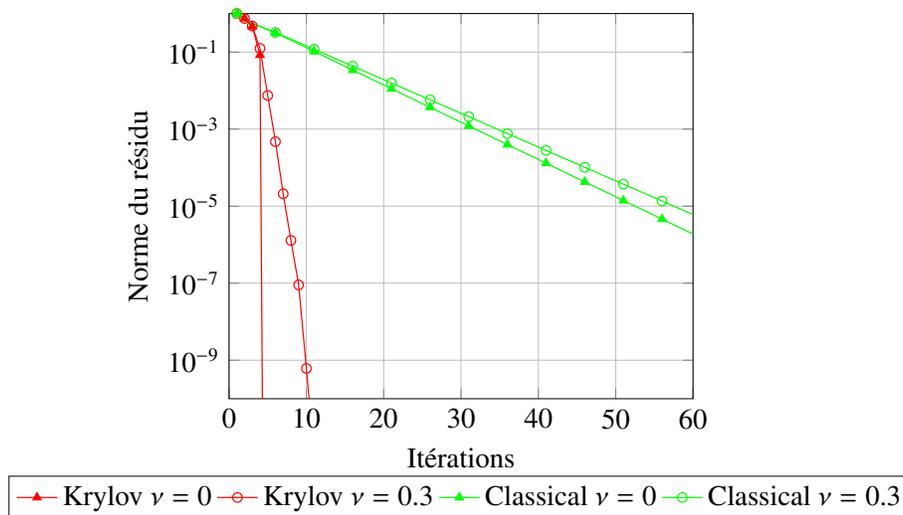


FIGURE 3.22 – Norme du résidu R_i

5.2.2 Cas 3D

Nous présentons avec cet exemple une comparaison 3D entre la méthode Latin avec relaxation classique et l’accélération de Krylov pour du mono et multi-échelle. Ces résultats sont comparés à un calcul éléments finis de référence sans sous-structuration.

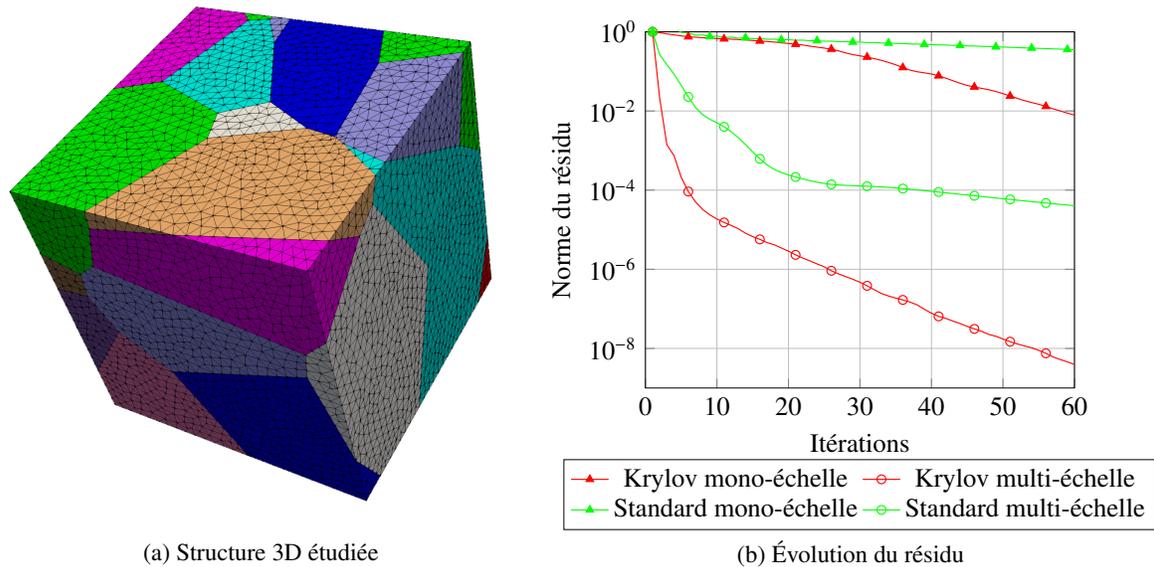


FIGURE 3.23 – Structure 3D étudiée

La structure étudiée est un cube sous-structuré en 32 sous-domaines pour lequel la face inférieure est encadrée et la face supérieure soumise à un effort de compression. La figure 3.23b montre l'évolution du résidu pour les implémentations mono et multi-échelles de l'approche classique et de l'approche avec accélération de Krylov.

Nous représentons aux figures 3.24 et 3.25 les cartes d'erreur normalisées sur le déplacement vertical obtenu après 60 itérations par rapport à un calcul direct de référence. Dans le cas de l'implémentation mono-échelle standard, l'écart par rapport à la référence est de l'ordre de 20% distribué sur toute la structure (Figure 3.24a). L'utilisation d'une accélération de Krylov, toujours dans un cadre mono-échelle, permet d'abaisser cet écart à 3% (Figure 3.24b). Bien entendu une approche multi-échelle permet d'améliorer la convergence. Rien que pour l'implémentation standard l'écart diminue à 0.7% (Figure 3.25a). L'accélération de Krylov couplé avec une approche multi-échelle amène à un écart moyen de 0.1% (Figure 3.25b).

Nous avons donc ici un gain net en terme de convergence. Nous pouvons cependant toujours visualiser des difficultés de convergence dans une zone d'interface multiple entre certains sous-domaines. Les arguments justifiant de telles difficultés ont pu être évoqués au sein du chapitre 2 à la section 3.3.

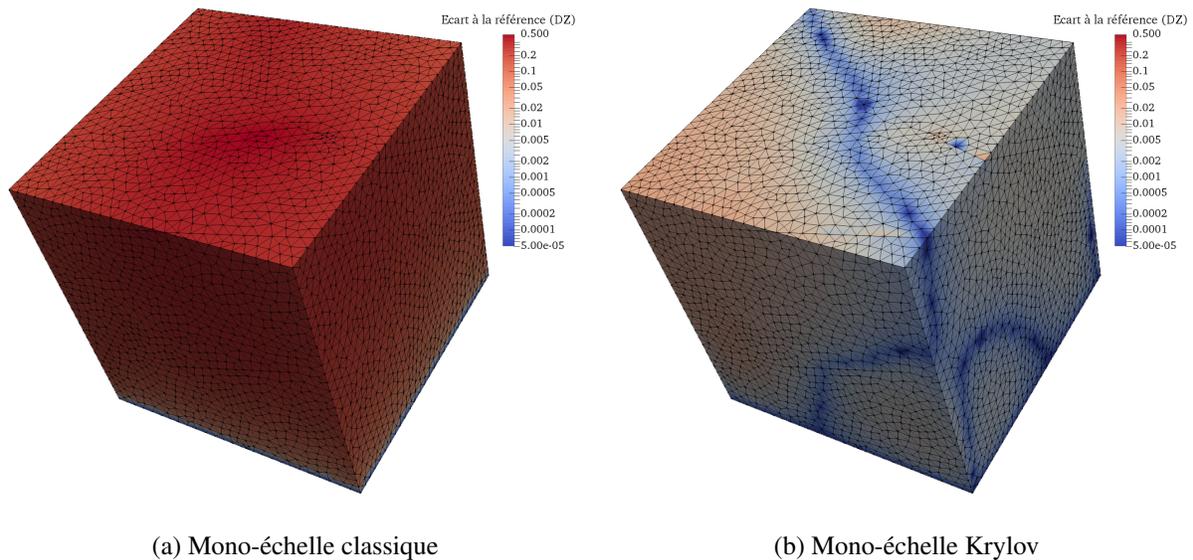


FIGURE 3.24 – Carte d'erreur normalisée par rapport à un calcul direct - mono-échelle

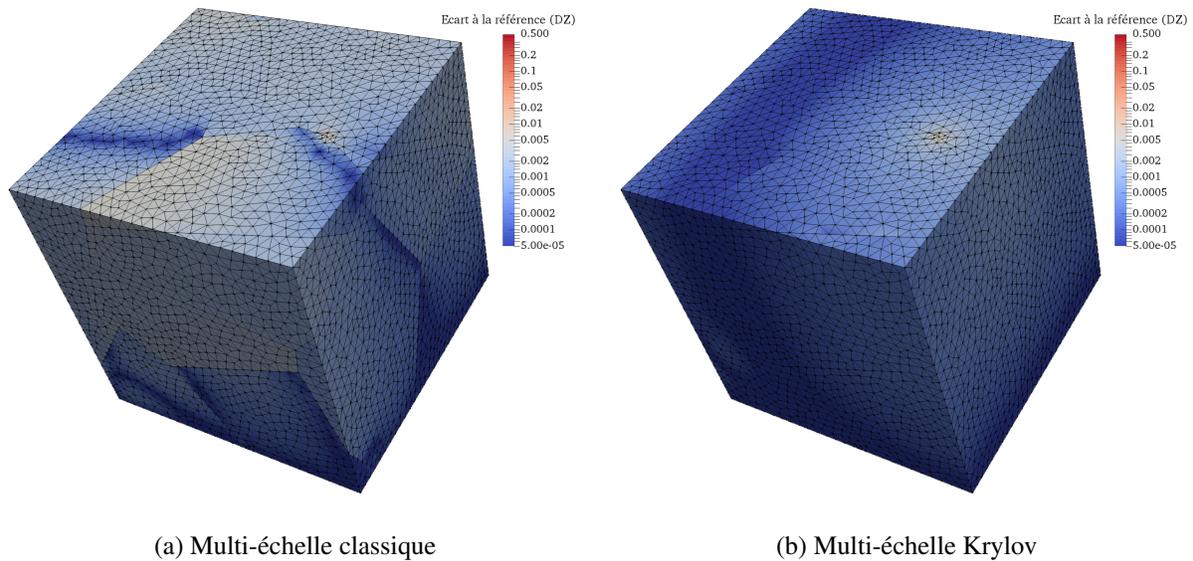


FIGURE 3.25 – Carte d'erreur normalisée par rapport à un calcul direct - multi-échelle

6 Bilan

Ce chapitre été l'objet du traitement de l'étape linéaire. L'extensibilité de la méthode a été obtenue grâce à une approche multi-échelle. Cette approche qui impose un équilibre global des efforts d'interface permet une propagation grossière des informations de conditions limites. Cette

approche déjà proposée par [Ladevèze et al., 2001, Nouy, 2003] a été re-écrite avec des notations algébriques. Cela nous a permis en particulier de faire apparaître le projecteur macroscopique associé au choix initial de base de déplacement.

Une réécriture de l'approche en continuité introduite dans [Loiseau, 2001] a été proposée. La continuité globale est prise en compte à l'aide d'un multiplicateur de Lagrange et un schéma à deux échelles permet de résoudre ce problème multi-échelle. Cependant, contrairement à l'approche en équilibre, il n'est pas possible d'interpréter l'approche en continuité comme une perturbation de la direction de recherche par un projecteur macroscopique. Le couplage entre les sous-domaines qui n'apparaît pas dans le cas de la continuité globale ne permet pas d'assurer l'extensibilité de la méthode.

Une étude de l'influence des directions de recherche sur la convergence permet de confirmer les résultats de [Loiseau, 2001]. L'introduction d'indicateurs séparant les contributions des déplacements d'interface des efforts d'interface permet de visualiser plus clairement l'influence de la direction de recherche. Nous avons aussi remis en cause le choix empirique du paramètre de relaxation. Dans le cadre de l'approche multi-échelle en équilibre appliquée à un problème simple, augmenter ce coefficient de relaxation ne permet pas d'accélérer la convergence. Une trop forte augmentation peut même s'accompagner de fortes instabilités et faire diverger la méthode.

La méthode Latin en décomposition de domaine pour des interfaces parfaites est équivalente à une méthode de lagrangien augmenté couplée à un algorithme de point fixe [Glowinski and Le Tallec, 1990] dont la convergence est réputée lente. Pour un problème d'élasticité linéaire avec des interfaces parfaites l'opérateur des itérés est linéaire et constant au cours des itérations. Nous avons donc proposé d'inclure ces itérations Latin au sein d'un solveur de Krylov afin d'accélérer la convergence. A une itération donnée la solution calculée est optimisée par rapport à l'ensemble des solutions précédemment déterminées. Nous pouvons noter sur des exemples 2D et 3D de fort gain en terme de convergence. Cependant pour des problèmes non-linéaires de type contact, l'opérateur des itérés n'est plus constant, et intégrer le solveur Latin dans un algorithme de Krylov devient plus complexe.

Formulation quasi-statique pour les non-linéarités de contact

Nous avons jusqu'ici présenté la méthode Latin dans un cadre statique. Cela a permis en particulier de simplifier les notations. Dans ce chapitre nous nous plaçons dans un cadre quasi-statique pour traiter des problèmes de contact frottant. L'hypothèse quasi-statique ne change en rien les différents principes de la méthode, mais nécessite l'introduction de nouvelles quantités, en particulier les vitesses d'interface. Nous présentons dans une première section les conséquences sur les différents algorithmes. Au sein d'une deuxième section nous nous intéressons à l'étape locale et plus particulièrement à la gestion d'interface de contact frottant. Une troisième section présente la prise en compte de précharge dans les assemblages à travers une légère modification des interfaces de type contact.

Sommaire

1	Méthode Latin dans un contexte quasi-statique	106
1.1	Mise en place du problème quasi-statique	107
1.1.1	Séparation des équations	107
1.1.2	Direction de recherche	107

1.2	Etape linéaire	108
1.2.1	Problème mono-échelle	108
1.2.2	Problème multi-échelle	108
1.3	Etape locale : interfaces parfaites	108
2	Etape locale : problème de contact aux interfaces	108
2.1	Formulation de l'étape locale pour un comportement de contact	108
2.2	Simplification de l'opérateur de direction de recherche non-locale	109
2.3	Résolution du problème de contact	111
2.3.1	Comportement normal et contact unilatéral	111
2.3.2	Frottement de Coulomb	112
2.3.3	Résumé de la résolution du problème de contact	115
2.4	Validation de l'implémentation des interfaces de contact	115
2.4.1	Comparaison avec un calcul Aster direct	115
2.4.2	Validation dans un cadre multi-paramétrique	118
3	Assemblages pré-chargés	124
3.1	Objectif de la pré-charge	124
3.2	Ecriture via une interface de contact	124
3.2.1	Choix de la sous-structuration pour la pré-charge	124
3.2.2	Conditions normales de l'étape locale de pré-charge	125
3.2.3	Conditions tangentielles de l'étape locale de pré-charge	126
3.3	Exemples numériques	126
3.3.1	Première validation du comportement	126
4	Application à un cas 3D	128
5	Bilan	129

1 Méthode Latin dans un contexte quasi-statique

Les problèmes de contact frottant nécessitent de pouvoir gérer une évolution. En effet, comme indiqué au chapitre 1 à la section 4, le frottement s'écrit à partir de la vitesse de glissement. Le cadre de la statique ne suffit plus, il est donc nécessaire de se placer sous l'hypothèse quasi-statique. De nouvelles grandeurs d'interface sont désormais introduites : $\widehat{\mathbf{W}}$ et \mathbf{W} respectivement les champs de vitesse de l'étape locale et de l'étape linéaire.

1.1 Mise en place du problème quasi-statique

1.1.1 Séparation des équations

Le problème est maintenant écrit en vitesse. Les groupes de solutions partielles \mathcal{A} et \mathcal{L} portent désormais sur les champs de vitesse et d'effort et s'écrivent :

$$\begin{aligned} \mathcal{A} : (\mathbf{F}^t, \dot{\mathbf{W}}^t)_t \text{ solutions de } \forall t, & \begin{cases} \mathbf{K}\mathbf{u}^t = \mathbf{f}_d^t + \mathbf{N}^T \mathbf{F}^t \\ \dot{\mathbf{W}}^t = \mathbf{N}\dot{\mathbf{u}}^t \\ \dot{\mathbf{u}}^t = (\mathbf{u}^t - \mathbf{u}^{t-\Delta t}) / \Delta t, + \text{conditions initiales} \end{cases} \\ \mathcal{L} : (\widehat{\mathbf{F}}^t, \widehat{\mathbf{W}}^t)_t \text{ solutions de } \forall t, & \widehat{\mathbf{F}}^t = \mathbf{B}^t \mathbf{b} \left(\mathbf{B}\widehat{\mathbf{W}}^t, t' \leq t, \mathbf{B}\widehat{\mathbf{W}}^0 \text{ donné} \right) \end{aligned} \quad (4.1)$$

Une nouvelle relation est ajoutée pour traduire le lien entre le déplacement et la vitesse. Un schéma d'intégration implicite est utilisé en initialisant par $\widehat{\mathbf{W}}^0$ donné. Les relations d'interface sont écrites de manière générale en introduisant \mathbf{b} la loi de comportement de l'interface.

Remarque 4.0

Dans le cas d'interface parfaite, cette loi s'écrit en déplacement :

$$\forall t, \begin{cases} \mathbf{A}\widehat{\mathbf{F}}^t = 0 \\ \mathbf{B}\widehat{\mathbf{W}}^t = 0 \end{cases} \quad (4.2)$$

1.1.2 Direction de recherche

La dualité effort/vitesse est naturelle et associée à la notion de puissance. Ainsi il est aussi naturel d'écrire les directions de recherche en vitesse. La direction de montée devient :

$$\forall t, \widehat{\mathbf{F}}^t - \mathbf{F}^t - \mathbf{k}^+ (\widehat{\mathbf{W}}^t - \dot{\mathbf{W}}^t) = 0 \quad (4.3)$$

La direction de descente s'écrit quant à elle :

$$\forall t, \mathbf{F}^t - \widehat{\mathbf{F}}^t + \mathbf{k}^- (\dot{\mathbf{W}}^t - \widehat{\mathbf{W}}^t) = 0 \quad (4.4)$$

Il faut désormais noter que \mathbf{k}^- et \mathbf{k}^+ relient non plus des champs de déplacement à des réactions nodales mais des champs de vitesse à des réactions nodales.

1.2 Etape linéaire

1.2.1 Problème mono-échelle

L'utilisation d'une direction de recherche en vitesse couplée avec le schéma d'intégration conduit à résoudre les problèmes mono-échelles indépendants par sous-domaine suivants :

$$\begin{aligned} &\text{Trouver } \mathbf{u}^{t+\Delta t} \text{ tel que } \forall t, \left(\mathbf{K} + \tilde{\mathbf{k}}/\Delta t \right) \mathbf{u}^{t+\Delta t} = \mathbf{f}_d^{t+\Delta t} + \mathbf{N}^T \left(\mathbf{F}^{t+\Delta t} + \mathbf{k}^- \dot{\mathbf{W}}^{t+\Delta t} + \frac{\mathbf{k}^-}{\Delta t} \mathbf{W}^t \right) \\ &\text{Ensuite : } \begin{cases} \dot{\mathbf{W}}^{t+\Delta t} = \mathbf{N} \dot{\mathbf{u}}^{t+\Delta t} = \mathbf{N} \left(\mathbf{u}^{t+\Delta t} - \mathbf{u}^t \right) / \Delta t \\ \mathbf{F}^{t+\Delta t} = \widehat{\mathbf{F}}^{t+\Delta t} + \mathbf{k}^- \left(\widehat{\mathbf{W}}^{t+\Delta t} - \dot{\mathbf{W}}^{t+\Delta t} \right) \end{cases} \end{aligned} \quad (4.5)$$

1.2.2 Problème multi-échelle

Nous nous focalisons sur l'approche multi-échelle en équilibre dont le principe est identique à l'approche en statique. Tout d'abord une première contribution au champ de déplacement est obtenue par le problème mono-échelle :

$$\text{Trouver } \mathbf{u}_1^{t+\Delta t} \text{ tel que } \forall t, \left(\mathbf{K} + \tilde{\mathbf{k}}/\Delta t \right) \mathbf{u}_1^{t+\Delta t} = \mathbf{f}_d^{t+\Delta t} + \mathbf{N}^T \left(\mathbf{F}^{t+\Delta t} + \mathbf{k}^- \dot{\mathbf{W}}^{t+\Delta t} + \frac{\mathbf{k}^-}{\Delta t} \mathbf{W}^t \right) \quad (4.6)$$

La seconde étape consiste à déterminer $\mathbf{u}_2^{t+\Delta t}$ tel que :

$$\mathbf{u}_2^{t+\Delta t} = - \left(\mathbf{K} + \tilde{\mathbf{k}}/\Delta t \right)^{-1} \mathbf{N}^T \mathbf{k}^- \mathbf{A}^T \mathbb{W} \mathbf{K}_M^{-1} \mathbb{W} \mathbf{A} \mathbf{k}^- \left[\left(\widehat{\mathbf{W}}^{t+\Delta t} + \frac{\mathbf{W}^t}{\Delta t} \right) - \frac{\mathbf{N} \mathbf{u}_1^{t+\Delta t}}{\Delta t} \right] \quad (4.7)$$

1.3 Etape locale : interfaces parfaites

Suivant le même principe que sous l'hypothèse statique, le problème à résoudre lors de l'étape locale quasi-statique pour des interfaces parfaites est :

$$\begin{aligned} &\text{Trouver } \widehat{\mathbf{W}}^{t+\Delta t} \text{ tel que } \forall t, \widehat{\mathbf{W}}^{t+\Delta t} = \mathbf{A}^T \left(\mathbf{A} \frac{\mathbf{k}^+}{\Delta t} \mathbf{A}^T \right) \left[\mathbf{A} \mathbf{k}^+ \dot{\mathbf{W}}^t + \mathbf{A} \frac{\mathbf{k}^+}{\Delta t} \widehat{\mathbf{W}}^t - \mathbf{A} \mathbf{F}^t \right] \\ &\text{Ensuite : } \begin{cases} \widehat{\mathbf{W}}^{t+\Delta t} = \left(\widehat{\mathbf{W}}^{t+\Delta t} - \widehat{\mathbf{W}}^t \right) / \Delta t \\ \mathbf{F}^{t+\Delta t} = \mathbf{F}^t + \mathbf{k}^+ \left(\widehat{\mathbf{W}}^{t+\Delta t} - \dot{\mathbf{W}}^{t+\Delta t} \right) \end{cases} \end{aligned} \quad (4.8)$$

2 Etape locale : problème de contact aux interfaces

2.1 Formulation de l'étape locale pour un comportement de contact

Pour écrire le problème de contact nous supposons que les interfaces concernées sont suffisamment régulières pour pouvoir définir à chaque noeud d'interface une base locale : n la normale et τ

pour la direction tangentielle. En considérant une interface Γ_{ij} avec $i < j$, la normale n est définie de i vers j . En 3D, la direction tangente est $\tau = (\tau_1, \tau_2)$. Nous faisons l'hypothèse que les grandeurs d'interface sont exprimées dans la base locale (n, τ) .

L'étape locale pour des interfaces de contact regroupe les différentes relations gouvernant les lois de contact, ainsi que la relation de direction de recherche écrite en vitesse. On ajoute un schéma d'intégration pour faire le lien entre les champs de vitesse et les champs de déplacement. Ainsi le problème à résoudre lors de l'étape locale s'écrit :

$$\forall t, \text{ Trouver } (\widehat{\mathbf{W}}^{t+\Delta t}, \widehat{\mathbf{F}}^{t+\Delta t}) \text{ solution de : } \begin{cases} \mathbf{B}_n \widehat{\mathbf{W}}_n^{t+\Delta t} + \mathbf{j}_n \geq 0 \\ \mathbf{A} \widehat{\mathbf{F}}^{t+\Delta t} = 0 \\ \mathbf{B}_n \widehat{\mathbf{F}}_n^{t+\Delta t} \geq 0 \\ (\mathbf{B}_n \widehat{\mathbf{F}}_n^{t+\Delta t})^T (\mathbf{B}_n \widehat{\mathbf{W}}_n^{t+\Delta t} + \mathbf{j}_n) = 0 \\ \widehat{\mathbf{F}}^{t+\Delta t} - \mathbf{F}^{t+\Delta t} - \mathbf{k}^+ \left(\widehat{\mathbf{W}}^{t+\Delta t} - \dot{\mathbf{W}}^{t+\Delta t} \right) = 0 \\ \widehat{\mathbf{W}}^{t+\Delta t} = \widehat{\mathbf{W}}^t + \Delta t \widehat{\dot{\mathbf{W}}}^{t+\Delta t}, \widehat{\mathbf{W}}^0 \text{ donné} \end{cases} \quad (4.9)$$

\mathbf{j}_n est le jeu initial normal à l'interface. Les quantités sans chapeau et les quantités à l'instant t sont connues respectivement lors de la précédente étape linéaire et le précédent pas de temps. Nous n'avons pas encore spécifié de conditions sur les parties tangentielles des efforts d'interface. Ces conditions distinguent le contact unilatéral du contact frottant.

Contact unilatéral : il est nécessaire de rajouter la condition :

$$\mathbf{B}_\tau \widehat{\mathbf{F}}_\tau^{t+\Delta t} = 0 \quad (4.10)$$

Il n'y a donc aucune transmission d'effort tangentiel à l'interface.

Contact frottant de Coulomb : on ajoute :

$$\begin{cases} \text{Si } \|\widehat{\mathbf{F}}_\tau^{t+\Delta t}\| < \mu |\widehat{\mathbf{F}}_n^{t+\Delta t}| \text{ alors } \mathbf{B}_\tau \widehat{\dot{\mathbf{W}}}_\tau^{t+\Delta t} = 0 & \text{adhérence} \\ \text{Si } \|\widehat{\mathbf{F}}_\tau^{t+\Delta t}\| = \mu |\widehat{\mathbf{F}}_n^{t+\Delta t}| \text{ alors } \exists \lambda > 0, \mathbf{B}_\tau \widehat{\dot{\mathbf{W}}}_\tau^{t+\Delta t} = -\lambda (\mathbf{B}_\tau \mathbf{B}_\tau^T)^{-1} \mathbf{B}_\tau \widehat{\mathbf{F}}_\tau^{t+\Delta t} & \text{glissement} \end{cases} \quad (4.11)$$

2.2 Simplification de l'opérateur de direction de recherche non-locale

Le choix non-intrusif de direction de recherche non-locale peut poser des difficultés pour résoudre le problème de contact à travers le couplage qu'introduiraient les semelles entre les noeuds d'interface. Pour éviter ces complications et conserver une solution explicite du problème de contact, il est nécessaire de simplifier la direction de recherche \mathbf{k}^+ afin de découpler les comportements des noeuds d'interface.

Cette simplification de l'opérateur de direction de recherche doit cependant respecter la dualité entre les champs de déplacement/vitesse et les champs d'effort correspondant aux réactions nodales

2.3 Résolution du problème de contact

2.3.1 Comportement normal et contact unilatéral

Nous nous intéressons dans un premier temps au cas du contact unilatéral (sans frottement). Nous étudions tout d'abord les conditions normales de contact aux interfaces afin de prédire les noeuds en contact.

Nous introduisons le schéma d'intégration dans la relation de non-interpénétration pour faire apparaître l'inconnue en vitesse $\widehat{\dot{W}}^{t+\Delta t}$:

$$\mathbf{B}_n \widehat{\dot{W}}_n^{t+\Delta t} + \mathbf{j}_n = \mathbf{B}_n \widehat{\dot{W}}_n^t + \mathbf{j}_n + \Delta t \mathbf{B}_n \widehat{\dot{W}}_n^{t+\Delta t} \geq 0 \quad (4.14)$$

Pour utiliser les conditions portant sur les efforts d'interface, nous introduisons la relation de direction de recherche dans la relation précédente :

$$\mathbf{B}_n \widehat{\dot{W}}_n^{t+\Delta t} + \mathbf{j}_n = \mathbf{B}_n \widehat{\dot{W}}_n^t + \mathbf{j}_n + \Delta t \mathbf{B}_n \left(\widehat{\dot{W}}_n^{t+\Delta t} + \mathbf{k}_n^{+1} \left(\widehat{\mathbf{F}}_n^{t+\Delta t} - \mathbf{F}_n^{t+\Delta t} \right) \right) \geq 0 \quad (4.15)$$

Cette expression peut être réorganisée pour faire apparaître les quantités connues d'une part et la grandeur d'effort inconnue $\widehat{\mathbf{F}}^{t+\Delta t}$ de l'autre part :

$$\underbrace{\mathbf{B}_n \widehat{\dot{W}}_n^t + \mathbf{j}_n + \Delta t \mathbf{B}_n \widehat{\dot{W}}_n^{t+\Delta t} - \Delta t \mathbf{B}_n \mathbf{k}_n^{+1} \mathbf{F}_n^{t+\Delta t}}_{C_n} + \Delta t \mathbf{B}_n \mathbf{k}_n^{+1} \widehat{\mathbf{F}}_n^{t+\Delta t} \geq 0 \quad (4.16)$$

C_n est une quantité déterminable complètement via des grandeurs connues lors de l'étape locale au pas de temps $t + \Delta t$. Cet indicateur nous permet d'évaluer noeud à noeud le statut de contact ou décollement. Puisque sur une même interface des noeuds peuvent être en contact quand d'autres sont décollés, l'écriture ne peut rester concaténée et la résolution doit être menée noeud à noeud. C'est pourquoi les notations non grasses se réfèrent à un noeud donné.

Il est alors possible de montrer une équivalence entre les statuts des noeuds en contact et la valeur du prédicteur C_n écrit pour un noeud d'interface :

Contact	\Leftrightarrow	$C_n \leq 0$
Décollement	\Leftrightarrow	$C_n > 0$

(4.17)

La démonstration de l'équivalence est en annexe 1.1. Nous exhibons maintenant les solutions normales explicites de contact.

Décollement : $C_n > 0$

Dans ce cas $\widehat{\mathbf{F}}_n^{t+\Delta t} = 0$ et $\widehat{\mathbf{F}}_\tau^{t+\Delta t} = 0$. Les vitesses et déplacements d'interface sont déterminés via la relation de direction de recherche et le schéma d'intégration :

$\widehat{\mathbf{F}}_n^{t+\Delta t}$	$= 0$
$\widehat{\dot{W}}_n^{t+\Delta t}$	$= \dot{W}_n^{t+\Delta t} - k_n^{+1} \mathbf{F}_n^{t+\Delta t}$
$\widehat{W}_n^{t+\Delta t}$	$= \widehat{W}_n^t + \Delta t \widehat{\dot{W}}_n^{t+\Delta t}$

(4.18)

Contact : $C_n \leq 0$ Dans le cas du contact il est remarqué que $\Delta t B_n k_n^{+1} \widehat{F}_n^{t+\Delta t} = -C_n$. Nous rappelons que $\widehat{F}_n^{t+\Delta t}$ peut aussi s'écrire sous la forme $\widehat{F}_n^{t+\Delta t} = B_n^T F_n^*$. Ainsi l'expression de la composante normale de l'effort est :

$$\widehat{F}_n^{t+\Delta t} = -B_n^T (B_n k_n^{+1} B_n^T)^{-1} \frac{C_n}{\Delta t} \quad (4.19)$$

Dans le cas de comportement unilatéral, les composantes tangentielles de l'effort sont nulles : $\widehat{F}_\tau^{t+\Delta t} = 0$. Les vitesses et déplacements sont obtenus via la relation de direction de recherche et le schéma d'intégration :

$$\begin{array}{l} \widehat{F}_n^{t+\Delta t} = -B_n^T (B_n k_n^{+1} B_n^T)^{-1} \frac{C_n}{\Delta t} \\ \widehat{F}_\tau^{t+\Delta t} = 0 \\ \widehat{W}_n^{t+\Delta t} = \dot{W}_n^{t+\Delta t} + k_n^{+1} (\widehat{F}_n^{t+\Delta t} - F_n^{t+\Delta t}) \\ \widehat{W}_\tau^{t+\Delta t} = \dot{W}_\tau^{t+\Delta t} + k_\tau^{+1} F_\tau^{t+\Delta t} \\ \widehat{W}^{t+\Delta t} = \widehat{W}^t + \Delta t \widehat{W}^{t+\Delta t} \end{array} \quad (4.20)$$

2.3.2 Frottement de Coulomb

Les conditions de frottement ne sont traitées que lorsque le statut de contact du noeud à été déterminé et que les grandeurs normales ont été calculées. Le cas sans frottement a été détaillé précédemment. Nous nous intéressons ici au frottement de Coulomb qui s'écrit noeud à noeud :

$$\begin{array}{l} \text{Si } \|\widehat{F}_\tau^{t+\Delta t}\| < \mu |\widehat{F}_n^{t+\Delta t}| \text{ alors } B_\tau \widehat{W}_\tau^{t+\Delta t} = 0 \quad \text{Adhérence} \\ \text{Si } \|\widehat{F}_\tau^{t+\Delta t}\| = \mu |\widehat{F}_n^{t+\Delta t}| \text{ alors } \exists \lambda > 0, B_\tau \widehat{W}_\tau^{t+\Delta t} = -\lambda (B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_\tau^{t+\Delta t} \quad \text{Glissement} \end{array} \quad (4.21)$$

en notant μ le coefficient de frottement de Coulomb. Nous rappelons que les efforts tangentiels doivent toujours être équilibrés : $A_\tau \widehat{F}_\tau^{t+\Delta t} = 0$.

Grâce au caractère défini positif de l'opérateur de direction de recherche k^+ , il est possible de décomposer le champ de vitesse sous la forme :

$$\exists (\alpha, \beta), \widehat{W}_\tau^{t+\Delta t} = A_\tau^T \alpha + k_\tau^{+1} B_\tau^T \beta \quad (4.22)$$

Remarque 4.1

$A_\tau^T \alpha$ correspond à la partie adhérente alors que $k_\tau^{+1} B_\tau^T \beta$ se réfère à la partie en glissement.

En injectant cette expression du champ de vitesse au sein de la relation de direction de recherche, on obtient :

$$\widehat{F}_\tau^{t+\Delta t} - F_\tau^{t+\Delta t} - k_\tau^+ A_\tau^T \alpha - B_\tau^T \beta + k_\tau^+ \dot{W}_\tau^{t+\Delta t} = 0 \quad (4.23)$$

En appliquant l'opérateur A_τ à cette relation, l'équilibre des efforts d'interface apparaît, faisant disparaître l'inconnue $\widehat{F}_\tau^{t+\Delta t}$. Ainsi il est possible d'exprimer α en fonction de quantités connues :

$$\alpha = (A_\tau k_\tau^+ A_\tau^T)^{-1} A_\tau^T [k_\tau^+ \dot{W}_\tau^{t+\Delta t} - F_\tau^{t+\Delta t}] \quad (4.24)$$

Remarque 4.2

La partie du déplacement relative à l'adhérence s'interprète comme une condition d'interface parfaite sur les composantes tangentielles.

L'expression de α est ensuite insérée dans la relation de direction de recherche pour exprimer $\widehat{F}_\tau^{t+\Delta t}$ en fonction de la partie relative au glissement $B_\tau^T \beta$.

$$\widehat{F}_\tau^{t+\Delta t} = \underbrace{\left[\mathbb{I} - k_\tau^+ A_\tau^T (A_\tau k_\tau^+ A_\tau^T)^{-1} A_\tau \right] F_\tau^{t+\Delta t} - k_\tau^+ \left[\mathbb{I} - A_\tau^T (A_\tau k_\tau^+ A_\tau^T)^{-1} A_\tau k_\tau^+ \right] \dot{W}_\tau^{t+\Delta t} + B_\tau^T \beta}_{G_\tau} \quad (4.25)$$

G_τ représente la contribution d'adhérence de l'effort d'interface. Il ne dépend que de grandeurs connues depuis l'étape linéaire et peut donc être directement calculé lors de l'étape locale. Cet indicateur nous informe sur l'état de glissement au niveau du noeud :

$$\begin{aligned} \text{Adhérence} &\Leftrightarrow \|G_\tau\| \leq \mu |\widehat{F}_n^{t+\Delta t}| \\ \text{Glissement} &\Leftrightarrow \|G_\tau\| > \mu |\widehat{F}_n^{t+\Delta t}| \end{aligned} \quad (4.26)$$

Remarque 4.3

La démonstration des équivalences est présentée à l'annexe 1.2.

Adhérence : $\|G_\tau\| \leq \mu |\widehat{F}_n^{t+\Delta t}|$

Dans ce cas, il est possible de poser directement $\beta = 0$ et ainsi $\widehat{F}_\tau^{t+\Delta t} = G_\tau$. La relation de direction de recherche est ensuite utilisée pour déterminer le champ de vitesse tangentielle.

$$\begin{aligned} \widehat{F}_\tau^{t+\Delta t} &= G_\tau \\ \widehat{W}_\tau^{t+\Delta t} &= \dot{W}_\tau^{t+\Delta t} + k_\tau^{+1} (\widehat{F}_\tau^{t+\Delta t} - F_\tau^{t+\Delta t}) \\ \widehat{W}_\tau^{t+\Delta t} &= \widehat{W}_\tau^t + \Delta t \widehat{W}_\tau^{t+\Delta t} \end{aligned} \quad (4.27)$$

Dans le cas de l'adhérence, le champ de vitesse peut aussi être directement calculé par :

$$\widehat{W}_\tau^{t+\Delta t} = A^T \tau (A_\tau k_\tau^+ A_\tau^T)^{-1} A_\tau^T [k_\tau^+ \dot{W}_\tau^{t+\Delta t} - F_\tau^{t+\Delta t}] \quad (4.28)$$

Glissement : $\|G_\tau\| > \mu|\widehat{F}_n^{t+\Delta t}|$

Pour du glissement, les deux inconnues β et γ doivent être déterminées telles que :

$$\begin{aligned} B_\tau \widehat{W}_\tau^{t+\Delta t} &= -\lambda (B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_\tau^{t+\Delta t} \\ \|\widehat{F}_\tau^{t+\Delta t}\| &= \mu|\widehat{F}_n^{t+\Delta t}| \end{aligned} \quad (4.29)$$

Suivant l'expression générale de la vitesse tangentielle à l'équation 4.22, $B_\tau \widehat{W}_\tau^{t+\Delta t}$ peut s'écrire :

$$B_\tau \widehat{W}_\tau^{t+\Delta t} = B_\tau k_\tau^{+-1} B_\tau^T \beta \quad (4.30)$$

Ainsi il est possible d'exprimer le champ d'effort tangentiel en fonction de β et λ :

$$(B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_\tau^{t+\Delta t} = - (B_\tau k_\tau^{+-1} B_\tau^T) \frac{\beta}{\lambda} \quad (4.31)$$

De plus nous rappelons que $\widehat{F}_\tau^{t+\Delta t} = G_\tau + B_\tau^T \beta$. Ainsi la relation entre β et γ est :

$$(B_\tau B_\tau^T)^{-1} B_\tau G_\tau + \beta = - (B_\tau k_\tau^{+-1} B_\tau^T) \frac{\beta}{\lambda} \quad (4.32)$$

Soit :

$$\beta = - \left(\frac{B_\tau k_\tau^{+-1} B_\tau^T}{\lambda} + \mathbb{I} \right)^{-1} (B_\tau B_\tau^T)^{-1} B_\tau G_\tau \quad (4.33)$$

L'effort tangentiel peut alors s'exprimer comme une fonction de l'inconnue λ et de la quantité connue G_τ :

$$\widehat{F}_\tau^{t+\Delta t} = \left(\mathbb{I} - B_\tau^T \left(\mathbb{I} + \frac{B_\tau k_\tau^{+-1} B_\tau^T}{\lambda} \right)^{-1} (B_\tau B_\tau^T)^{-1} B_\tau \right) G_\tau \quad (4.34)$$

L'inconnue λ est déterminée par la relation $\|\widehat{F}_\tau^{t+\Delta t}\| = \mu|\widehat{F}_n^{t+\Delta t}|$ imposée par la condition de glissement.

Nous considérons pour cela la quantité intermédiaire $\|(B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_\tau^{t+\Delta t}\| = \mu|(B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_n^{t+\Delta t}|$ qui n'est pas redondante de chaque côté de l'interface. Le problème devient alors :

$$\left\| \left(\mathbb{I} - \left(\mathbb{I} + \frac{B_\tau k_\tau^{+-1} B_\tau^T}{\lambda} \right)^{-1} \right) \underbrace{(B_\tau B_\tau^T)^{-1} B_\tau G_\tau}_{G'_\tau} \right\| = \underbrace{\mu|(B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_n^{t+\Delta t}|}_D \quad (4.35)$$

Soit :

$$\left\| \left(\mathbb{I} - \left(\mathbb{I} + \frac{B_\tau k_\tau^{+-1} B_\tau^T}{\lambda} \right)^{-1} \right) G'_\tau \right\| = D \quad (4.36)$$

Il faut noter que l'opérateur appliqué à G'_τ est diagonal ce qui va faciliter la résolution du problème. Ainsi l'effort tangentiel peut s'exprimer en fonction de G'_τ comme suit :

$$\widehat{F}_\tau^{t+\Delta t} = \underline{\underline{\gamma}} G'_\tau \quad (4.37)$$

$$\text{avec } \underline{\underline{\gamma}} = \left(\mathbb{I} - \left(\mathbb{I} + \frac{B_\tau k_\tau^{+1} B_\tau^T}{\lambda} \right)^{-1} \right).$$

Nous distinguons le cas d'un problème 2D de celui 3D :

— en 2D l'effort tangentiel ne comporte qu'une seule composante et ainsi $\underline{\underline{\gamma}} \in \mathbb{R}$ et $G'_\tau \neq 0$.

L'inconnue $\underline{\underline{\gamma}}$ est alors calculée aisément par $\underline{\underline{\gamma}} = \frac{\mu |\widehat{F}_n^{t+\Delta t}|}{\|G'_\tau\|}$ et ainsi la composante tangentielle de l'effort est :

$$\widehat{F}_\tau^{t+\Delta t} = \mu |\widehat{F}_n^{t+\Delta t}| \frac{G'_\tau}{\|G'_\tau\|} \quad (4.38)$$

— en 3D l'équation 4.36 est une équation polynomiale de degré 4. Cependant dans le cas où les directions de recherche tangentielles sont égales : $k_{\tau_1}^+ = k_{\tau_2}^+$, $\underline{\underline{\gamma}} = \gamma \mathbb{I}$ et une solution similaire au cas 2D peut être calculée :

$$\widehat{F}_\tau^{t+\Delta t} = \mu |\widehat{F}_n^{t+\Delta t}| \frac{G'_\tau}{\|G'_\tau\|} \quad (4.39)$$

Les vitesses et déplacements d'interface sont ensuite déterminées via la relation de direction de recherche et le schéma implicite.

$$\boxed{\begin{aligned} \widehat{F}_\tau^{t+\Delta t} &= \mu |\widehat{F}_n^{t+\Delta t}| \frac{G'_\tau}{\|G'_\tau\|} \\ \widehat{W}_\tau^{t+\Delta t} &= \dot{W}_\tau^{t+\Delta t} + k_\tau^{+1} \left(\widehat{F}_\tau^{t+\Delta t} - F_\tau^{t+\Delta t} \right) \end{aligned}} \quad (4.40)$$

2.3.3 Résumé de la résolution du problème de contact

Nous résumons la résolution du problème local de contact à l'algorithme 2.1. Les indicateurs que nous avons présentés précédemment sont équivalents à ceux proposés classiquement dans la littérature [Blanze et al., 1996]. Cependant nous avons utilisé une écriture algébrique plus générale de ces indicateurs prenant en compte de potentielles différences des opérateurs de direction de recherche de chaque coté des interfaces.

2.4 Validation de l'implémentation des interfaces de contact

2.4.1 Comparaison avec un calcul Aster direct

Nous comparons l'implémentation des interfaces de contact de la méthode Latin non-intrusive par rapport à un calcul non-linéaire de contact effectué avec *code_aster* sans décomposition de domaine. La résolution du contact frottant dans *code_aster* est faite par une méthode de Newton avec lagrangien pour le contact et le frottement.

On considère une structure composée de deux sous-domaines S_0 et S_1 reliés par une interface Γ de contact frottant paramétrée par le coefficient de Coulomb μ . Six pas de chargement sont pris en compte. Dans un premier temps la structure est chargée par l'effort imposé F_0 afin d'introduire un effort normal à la zone de contact. Cet effort est maintenu pour l'ensemble des pas de chargement.

Algorithme 2.1 : Dans le cas d'une interface de contact

for Chaque noeud d'interface **do**

Calcul de l'indicateur de contact C_n

if $C_n > 0$ **then** Cas du décollement

$$\widehat{F}^{t+\Delta t} = 0$$

$$\widehat{W}^{t+\Delta t} = \dot{W}^{t+\Delta t} - k^{+1} F^{t+\Delta t} \quad (4.41)$$

$$\widehat{W}^{t+\Delta t} = \widehat{W}^t + \Delta t \widehat{W}^{t+\Delta t}$$

else if $C_n \leq 0$ **then** Cas du contact :

Calcul des composantes normales :

$$\widehat{F}_n^{t+\Delta t} = -B_n (B_n k_n^{+1} B_n^T)^{-1} \frac{C_n}{\Delta t} \quad (4.42)$$

$$\widehat{W}_n^{t+\Delta t} = \dot{W}_n^{t+\Delta t} + k_n^{+1} (\widehat{F}_n^{t+\Delta t} - F_n^{t+\Delta t})$$

if Contact unilatéral **then**

Effort tangentiel nul :

$$\widehat{F}_\tau^{t+\Delta t} = 0 \quad (4.43)$$

else if Contact frottant **then**

Calcul de l'indicateur de glissement G_τ

if $\|G_\tau\| \leq \mu |\widehat{F}_n^{t+\Delta t}|$ **then** Cas de l'adhérence

Calcul de l'effort tangentiel :

$$\widehat{F}_\tau^{t+\Delta t} = G_\tau \quad (4.44)$$

else if $\|G_\tau\| > \mu |\widehat{F}_n^{t+\Delta t}|$ **then** Cas du glissement

Calcul de l'effort tangentiel :

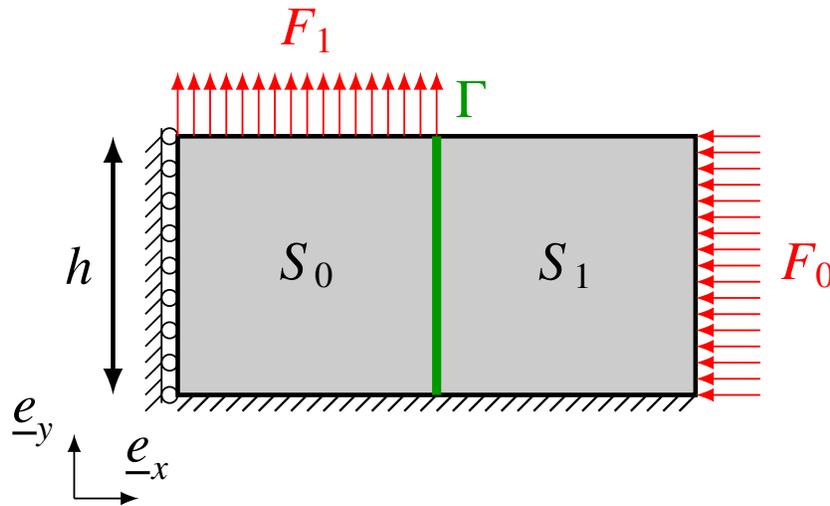
$$\widehat{F}_\tau^{t+\Delta t} = \mu |\widehat{F}_n^{t+\Delta t}| \frac{G'_\tau}{\|G'_\tau\|} \quad (4.45)$$

Calcul de la vitesse tangentielle via la direction de recherche :

$$\widehat{W}_\tau^{t+\Delta t} = \dot{W}_\tau^{t+\Delta t} + k_\tau^{+1} (\widehat{F}_\tau^{t+\Delta t} - F_\tau^{t+\Delta t}) \quad (4.46)$$

Schéma implicite pour déterminer le déplacement :

$$\widehat{W}^{t+\Delta t} = \widehat{W}^t + \Delta t \widehat{W}^{t+\Delta t} \quad (4.47)$$

FIGURE 4.2 – Cas test pour comparaison avec *code_aster*

Paramètres	Valeur
E	200 GPa
ν	0.3
h	10 mm
F_1^{max}	4 MPa
F_2	5 MPa
Δt	1s
critère d'erreur η_c	10^{-8}

TABLE 4.1 – Paramètres

Ensuite le sous-domaine S_0 subit progressivement un chargement F_1 jusqu'à F_1^{max} . La structure est maillée en élément quadrangle de taille 0.1mm.

L'évolution de l'indicateur d'erreur Latin est représentée à la figure 4.3.

Nous présentons aux figures 4.4 les champs de déplacement selon y pour les différents pas de chargement. Nous observons que pour les 3 premiers pas de chargement les deux sous-domaines adhèrent entre eux : l'effort F_1 n'est pas suffisant pour vaincre le frottement. Cependant à partir du quatrième pas de chargement une solution de type glissement apparaît à l'interface. Une discontinuité du déplacement tangentiel est observable.

La figure 4.5 montre les cartes d'erreur sur le champ de déplacement par rapport à un calcul de référence sous *code_aster*. Ces cartes nous indiquent une bonne correspondance de l'ordre de 0.05% avec des différences très localisées. En particulier le calcul *Aster* présente des difficultés pour traiter des interfaces de contact dont certains noeuds sont en commun avec une zone de conditions limites. La figure 4.6 illustre cela : un problème très local sur deux noeuds explique les différences entre le calcul *Latin* et le calcul *Aster*.

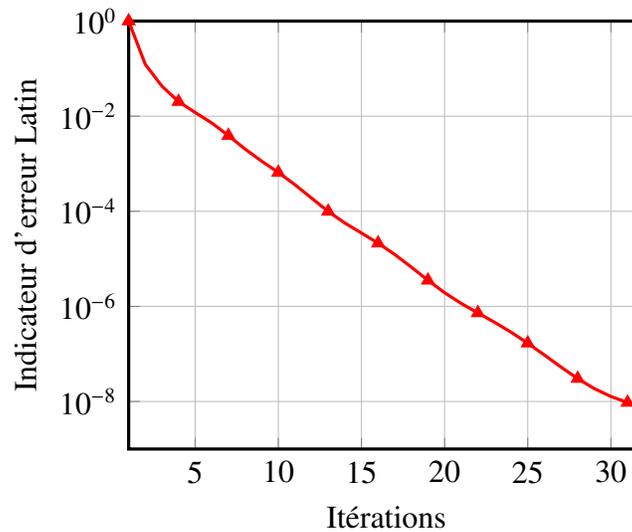


FIGURE 4.3 – Evolution de l'indicateur d'erreur Latin

Remarque 4.4

De façon générale les mouvements de corps rigides posent de vraies difficultés pour l'inversion des problèmes. Souvent ces mouvements ne sont bloqués que par des conditions de contact et de frottement. Les initialisations dans les codes industriels des statuts des noeuds se font par des statuts de glissement et font alors apparaître de nombreux mouvements de corps rigides.

2.4.2 Validation dans un cadre multi-paramétrique

La validation de l'implémentation des interfaces de contact est réalisée sur un cas test académique tiré de [Boucard and Champaney, 2003]. Le problème quasi-statique est présenté à la figure 4.7

La structure est constituée de 3 sous-domaines S_1, S_2 et S_3 . Deux interfaces de contact frottant lient respectivement (S_1, S_2) et (S_2, S_3) paramétrées par les coefficients de frottement μ_1 et μ_2 . Une interface supplémentaire permet de gérer une condition de bord de contact unilatéral avec jeu initial j sur le sous-domaine S_2 . Les matériaux sont élastiques linéaires isotropes de module de Young E et de coefficient de Poisson ν . Chaque sous-structure est discrétisée en 25×25 éléments de type quadrangle.

La structure est soumise à deux chargements F_1 et F_2 définis pour deux pas de temps t_0 et t_1 . Au premier temps à t_0 la structure est pré-chargée par F_1 afin d'imposer un effort de contact aux interfaces (S_1, S_2) et (S_2, S_3) . Au second pas de chargement, l'effort F_2 est appliqué à la sous-structure S_2 . L'objectif est d'étudier l'influence des coefficients de frottement sur le contact au niveau de l'interface avec jeu initial. Les coefficients de frottement varient entre 0 et 0.6. Nous choisissons 13 valeurs pour chaque coefficient. Les différents paramètres sont précisés au tableau 4.2.

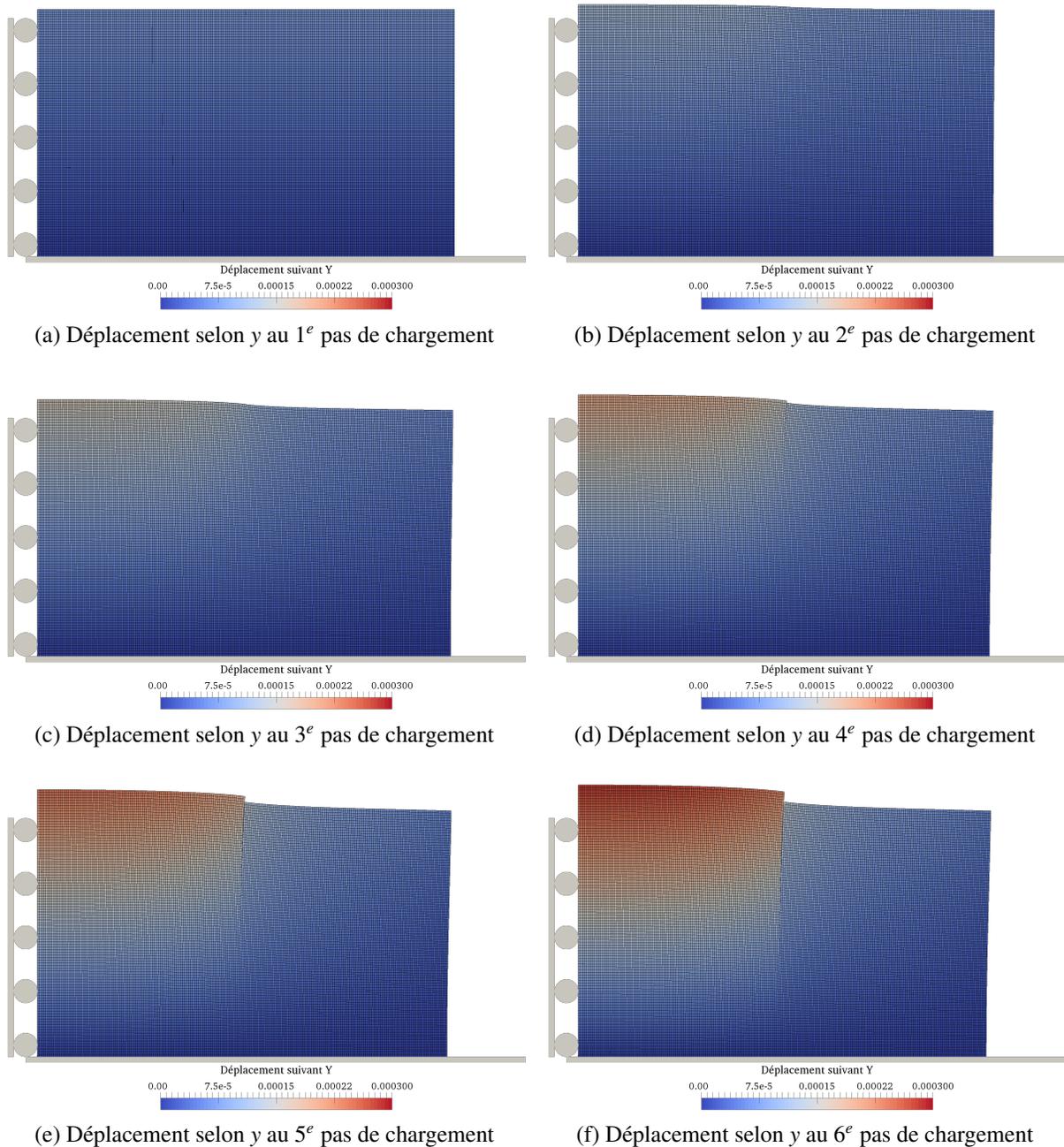
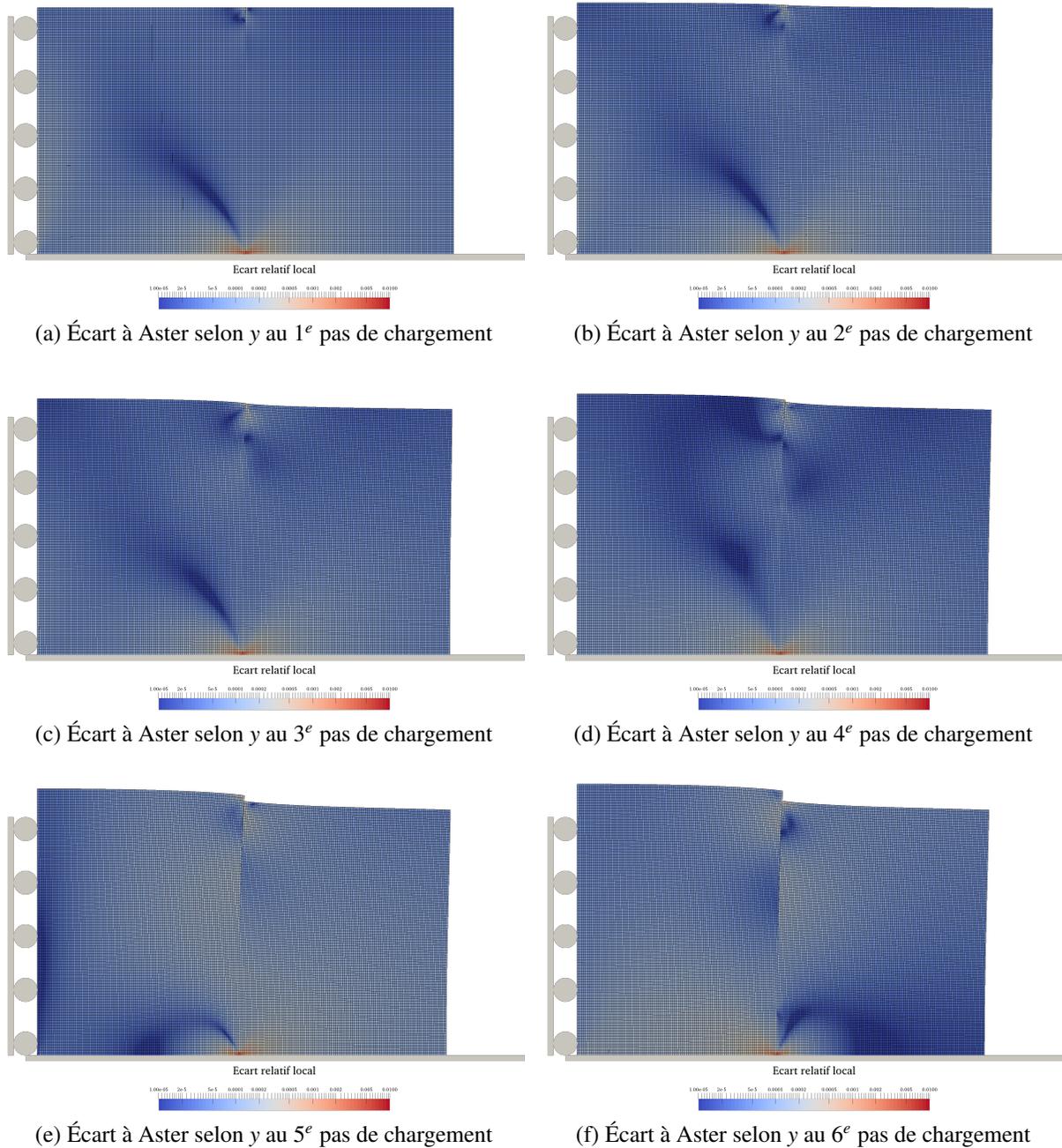


FIGURE 4.4 – Déformée ($\times 500$)

169 couples de coefficients de frottement (μ_1, μ_2) sont à étudier. C'est-à-dire potentiellement 169 calculs complets à effectuer. Nous mettons alors en place une approche multi-paramétrique dont le principe a été rappelé à la section 1.3. Pour un couple (μ_1, μ_2) donné, nous effectuons le calcul jusqu'à convergence $\eta < 10^{-4}$. Une fois le critère atteint, le calcul se poursuit en changeant

FIGURE 4.5 – Déformée ($\times 500$)

de couple (μ_1, μ_2) . Nous commençons par le couple $(\mu_1, \mu_2) = (0, 0)$. μ_1 est d'abord gardé fixe pendant que μ_2 est modifié à chaque convergence. Une fois les 13 valeurs de μ_2 étudiées, μ_1 est modifié à son tour. Cela se poursuit jusqu'à avoir balayé l'ensemble des couples (μ_1, μ_2) .

La figure 4.8a représente le calcul complet pour balayer les 169 couples (μ_1, μ_2) . 388 itéra-

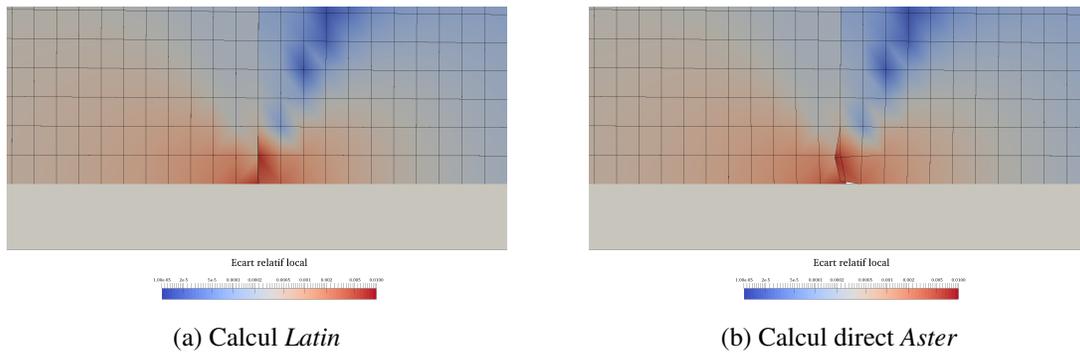


FIGURE 4.6 – Zoom au coin d'interface et conditions limites

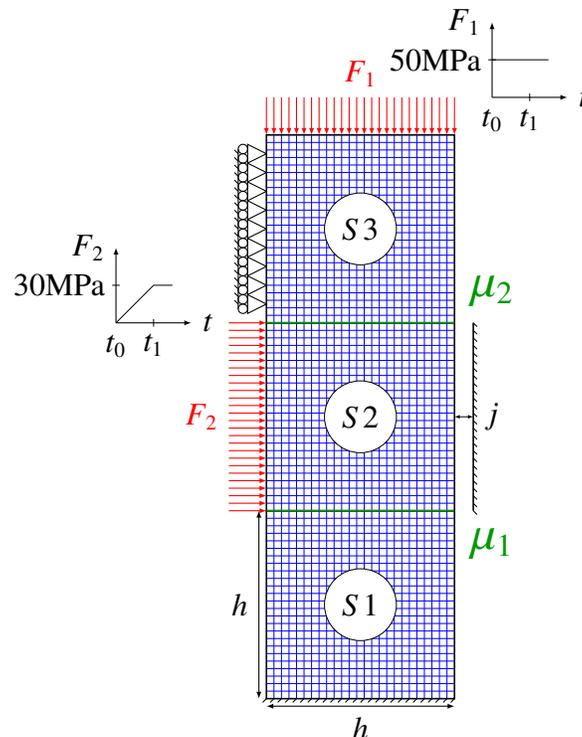


FIGURE 4.7 – Cas test 2D pour le contact frottant

tions ont été nécessaires pour obtenir ces 169 résultats. Chaque pic représente un changement de couple de paramètres. Ces remontées de l'indicateur d'erreur illustrent parfaitement l'intérêt de cette approche multi-paramétrique. En ne repartant pas d'une initialisation nulle, il est plus rapide d'atteindre le critère d'erreur objectif. A la figure 4.8b nous nous focalisons sur les premières itérations. Les 14 premières itérations correspondent au calcul pour le couple de paramètres (0,0). Cela permet d'évaluer grossièrement le nombre total d'itérations nécessaire pour obtenir l'ensemble des 169 résultats sans approche multi-paramétrique : $14 \times 169 = 2366$ itérations. Cela représente un gain de 6 sur le nombre d'itérations. Une fois le critère atteint pour le premier couple de paramètres, μ

Paramètres	Valeur
E	210 GPa
ν	0.3
h	50 mm
j	0.04 mm
F_1^{max}	50 MPa
F_2^{max}	30 MPa
μ_1, μ_2	$\in [0, 0.6]$
Δt	1 s
critère d'erreur η_c	10^{-4}

TABLE 4.2 – Paramètres

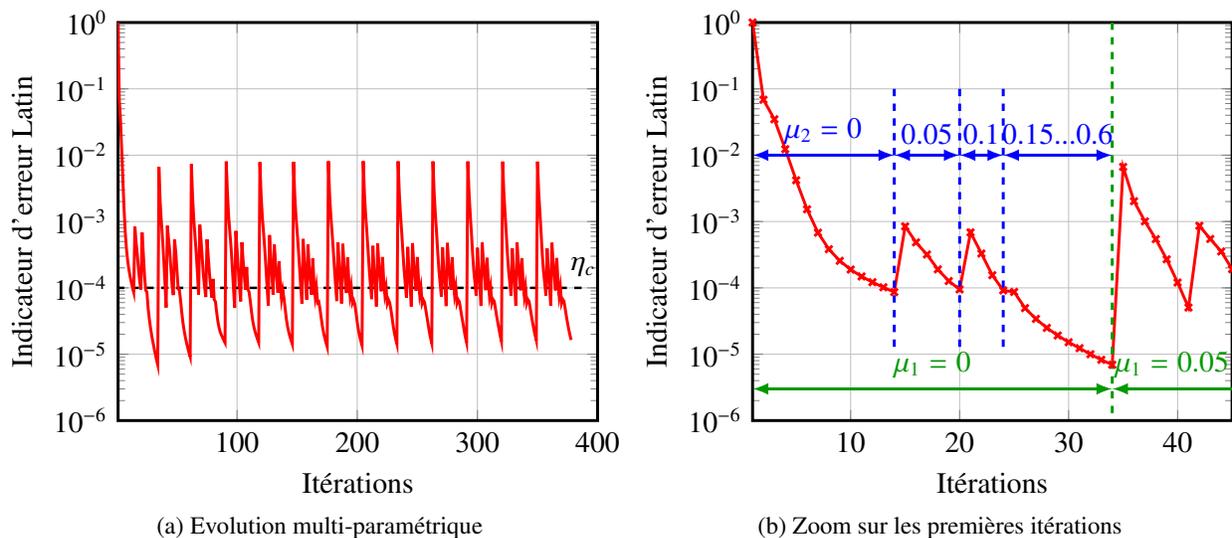


FIGURE 4.8 – Évolution de l'indicateur pour le calcul multi-paramétrique

est modifié de 0 à 0.05. Cela explique la remontée de l'indicateur d'erreur à la 15^e itération. Ensuite seulement 6 itérations sont nécessaires pour atteindre la convergence. A partir de la 24^e itération les initialisations permettent de converger en une seule itération. Les solutions sont tellement proches les unes des autres qu'il n'est pas nécessaire d'effectuer d'itération supplémentaire. A la 34^e itération l'ensemble des couples $(0, \mu_2)$ a été déterminé, μ_1 est donc à son tour modifié.

Suivant les coefficients choisis, le contact est potentiellement atteint partiellement ou totalement sur l'interface de contact avec jeu. Nous représentons alors la réaction à cette interface qui nous permet de quantifier à quel point le contact est atteint (Figure 4.9). Ces résultats sont en accord avec ceux de [Boucard and Champany, 2003] confirmant ainsi notre implémentation non-intrusive du contact. Nous pouvons en particulier distinguer quatre couples de paramètres pour illustrer les différents comportements de la structure. La figure 4.10a représente le cas particulier sans frottement. Ainsi quelle que soit la pré-charge F_1 imposée, le sous-domaine S_2 va se retrouver en contact sur

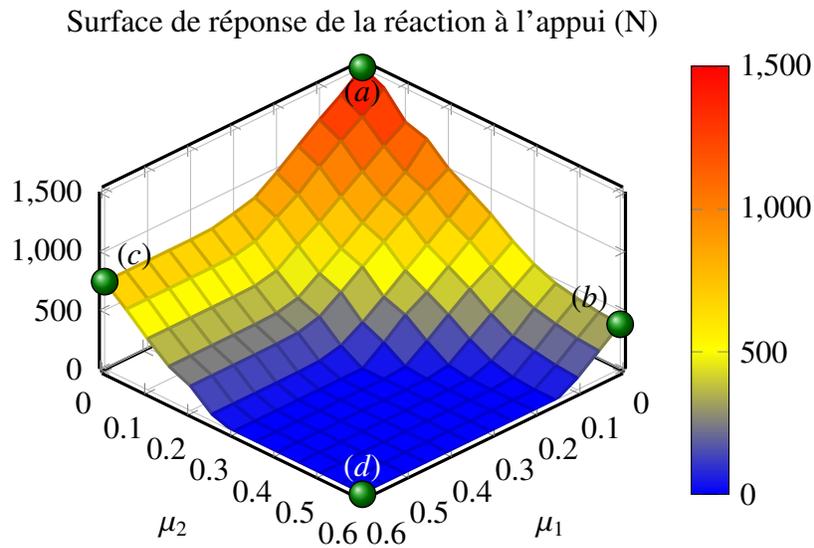


FIGURE 4.9 – Réaction au contact en fonction des coefficients de frottement

l'ensemble de l'interface avec jeu initial, transférant alors l'ensemble de la charge appliquée F_2 . Les figures 4.10b, 4.10c illustrent des cas pour lesquels le contact n'est atteint que partiellement. Le frottement à l'une des interfaces permet de "retenir" en partie le sous-domaine S_2 . La charge F_2 n'est donc pas intégralement transmise au niveau de l'interface avec jeu. A la figure 4.10d les coefficients de frottements des deux interfaces sont suffisamment importants pour retenir complètement le sous-domaine S_2 . Aucun contact n'est alors détecté à l'interface avec jeu.

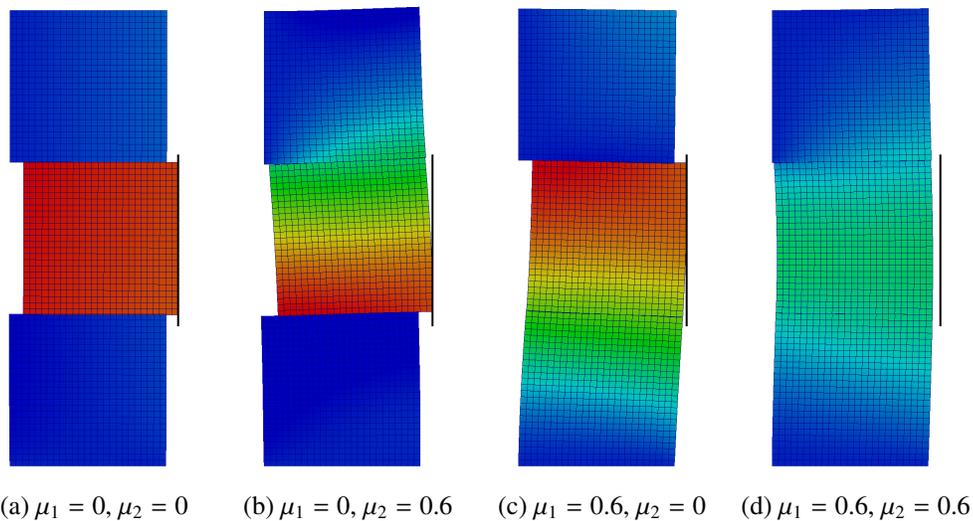


FIGURE 4.10 – Déformée au second pas de chargement - La carte de couleur correspond au déplacement horizontal.

3 Assemblages pré-chargés

3.1 Objectif de la pré-charge

La pré-charge de structures assemblées correspond à un état dit de référence pour lequel des contraintes non nulles sont imposées dans la structure. Les éléments d'assemblage de type vis sont soumis, en particulier, à un état de contrainte de tension afin d'imposer par réciprocity un état de compression dans les pièces adjacentes.

Cet état de contrainte est souvent obtenu en imposant un couple de serrage sur les écrous. Appliqué à l'aide de clé dynamométrique, seul le couple exercé sur l'écrou est maîtrisé. Cependant le frottement au niveau des éléments filetés ainsi que la rigidité locale des pièces ne permettent pas de contrôler de façon précise l'état de contrainte réellement imposé. Une variation d'environ plus ou moins 30% est couramment constatée sur la valeur de la pré-charge.

La pré-charge a pour but d'éviter de solliciter la vis en cisaillement. En effet en pré-chargeant, une contrainte normale de compression est appliquée entre les pièces serrées par le boulon. Ainsi, c'est l'adhérence entre ces deux pièces qui va permettre la transmission des efforts et éviter un glissement relatif. Tant qu'il n'y a pas de contact entre la tige de la vis et les pièces assemblées, la vis n'est pas cisailée. Cet état est directement lié à la valeur de la pré-charge imposée.

3.2 Ecriture via une interface de contact

3.2.1 Choix de la sous-structuration pour la pré-charge

L'objectif est d'imposer un état de contrainte initiale non nulle à la structure. Dans le cadre de la méthode Latin il est proposé dans [Champaney, 1996] d'imposer un déplacement relatif axial entre l'écrou et la tige de la vis (Figure 4.11). Cela permet de solliciter la vis en tension et d'appliquer une contrainte de compression aux pièces assemblées. Ce déplacement relatif est pris en compte à travers une interface de type contact pour laquelle un jeu axial initial négatif est appliqué.

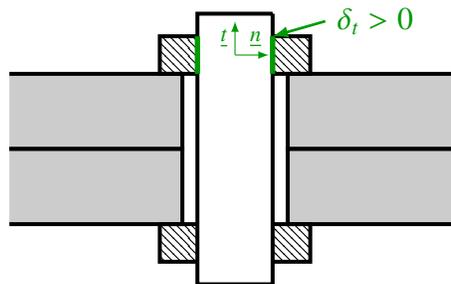


FIGURE 4.11 – Pré-charge orthoradiale

Avec *code_aster* les normales à une interface peuvent être déterminées très facilement. Cependant les directions tangentielles ne sont pas uniques. Lorsque les vis sont alignées avec un des axes canoniques $\underline{e}_x, \underline{e}_y$ ou encore \underline{e}_z , il est simple d'obtenir une direction tangentielle colinéaire avec l'axe de la tige de vis pour imposer un déplacement relatif entre celle-ci et l'écrou. Cependant il est plus

complexe de maîtriser l'orientation des directions tangentielles dans le cadre d'une implémentation générique. Nous choisissons donc de ne pas imposer la pré-charge par cette technique.

La seule contrainte que nous nous imposons est que la tige de la vis soit préalablement décomposée en deux sous-domaines dont l'interface entre les deux est plane et orthogonale à l'axe de la tige. La normale, est nécessairement alignée avec la tige, il devient alors plus simple de gérer de façon générique ce type d'interface. Le principe est alors d'imposer une interpénétration entre les deux sous-domaines au niveau de l'interface. Ainsi une sollicitation de tension est appliquée à ces deux parties de la tige (Figure 4.12). Les liaisons entre la tige et les écrous sont alors parfaites.

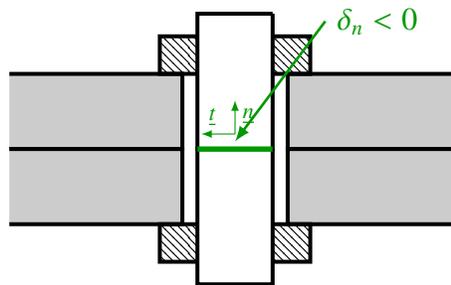


FIGURE 4.12 – Pré-charge normale

3.2.2 Conditions normales de l'étape locale de pré-charge

Cependant obtenir l'état de contrainte voulue en imposant un jeu négatif n'est pas simple et peu documenté dans la littérature. C'est pourquoi nous choisissons ici de détailler notre implémentation d'une interface de type pré-charge. Pour cela nous choisissons le cas de la figure 4.13. Deux sous-structures S_1 et S_2 sont considérées. Elles sont encastrées de part et d'autre de l'interface Γ . L'objectif est d'imposer une interpénétration \mathbf{p} des deux sous-domaines afin d'obtenir une contrainte normale de tension.

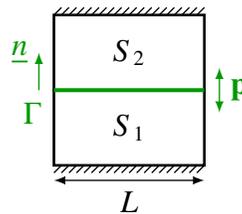


FIGURE 4.13 – Exemple de précharge

Cette pré-charge est gérée à l'étape locale au niveau de l'interface. En utilisant une interface de type contact, il est nécessaire de réécrire la relation de non-interpénétration $\mathbf{B}_n \widehat{\mathbf{W}}_n^{t+\Delta t} + \mathbf{j}_n \geq 0$. Désormais en notant $\mathbf{p} > 0$ la valeur de l'interpénétration associée à la pré-charge, la nouvelle condition pour imposer l'interpénétration est :

$$\mathbf{B}_n \widehat{\mathbf{W}}_n^{t+\Delta t} \leq -\mathbf{p} \quad (4.48)$$

Ou encore :

$$\mathbf{B}_n \widehat{\mathbf{W}}_n^{t+\Delta t} + \mathbf{p} \leq 0 \quad (4.49)$$

Cela permet d'adapter facilement le comportement de contact décrit à la section 2. En effet dans ce cas, il suffit d'inverser le signe de l'indicateur de contact pour ensuite utiliser le même schéma de résolution que celui décrit à l'algorithme 2.1.

Remarque 4.5

Ce renversement des conditions de contact impose de ne pas charger axialement les vis en compression. Puisque la condition a été inversée, une sollicitation de compression conduirait à une solution de type décollement à l'interface des deux parties de la tige. Une interpénétration artificielle plus grande que celle introduite par la pré-charge apparaîtrait.

3.2.3 Conditions tangentielles de l'étape locale de pré-charge

Les deux sous-domaines constituant la tige de la vis se réfèrent au même solide et des conditions d'interface parfaite doivent être imposées sur les grandeurs tangentielles. Ainsi en plus de garantir l'équilibre des efforts, la continuité des déplacements tangentiels à l'interface est vérifiée.

Ainsi les grandeurs tangentielles sont déterminées par :

$$\begin{aligned} \text{Trouver } \widehat{\mathbf{W}}_\tau^{t+\Delta t} \text{ tel que } \forall t, \widehat{\mathbf{W}}_\tau^{t+\Delta t} &= \mathbf{A}_\tau^T \left(\mathbf{A}_\tau \frac{\mathbf{k}^+_\tau}{\Delta t} \mathbf{A}_\tau^T \right) \left[\mathbf{A}_\tau \mathbf{k}^+_\tau \dot{\mathbf{W}}_\tau^t + \mathbf{A}_\tau \frac{\mathbf{k}^+_\tau}{\Delta t} \widehat{\mathbf{W}}_\tau^t - \mathbf{A}_\tau \mathbf{F}_\tau^t \right] \\ \text{Ensuite : } \begin{cases} \widehat{\dot{\mathbf{W}}}_\tau^{t+\Delta t} &= (\widehat{\mathbf{W}}_\tau^{t+\Delta t} - \widehat{\mathbf{W}}_\tau^t) / \Delta t \\ \mathbf{F}_\tau^{t+\Delta t} &= \mathbf{F}_\tau^t + \mathbf{k}^+_\tau \left(\widehat{\mathbf{W}}_\tau^{t+\Delta t} - \dot{\mathbf{W}}_\tau^{t+\Delta t} \right) \end{cases} \end{aligned} \quad (4.50)$$

3.3 Exemples numériques

3.3.1 Première validation du comportement

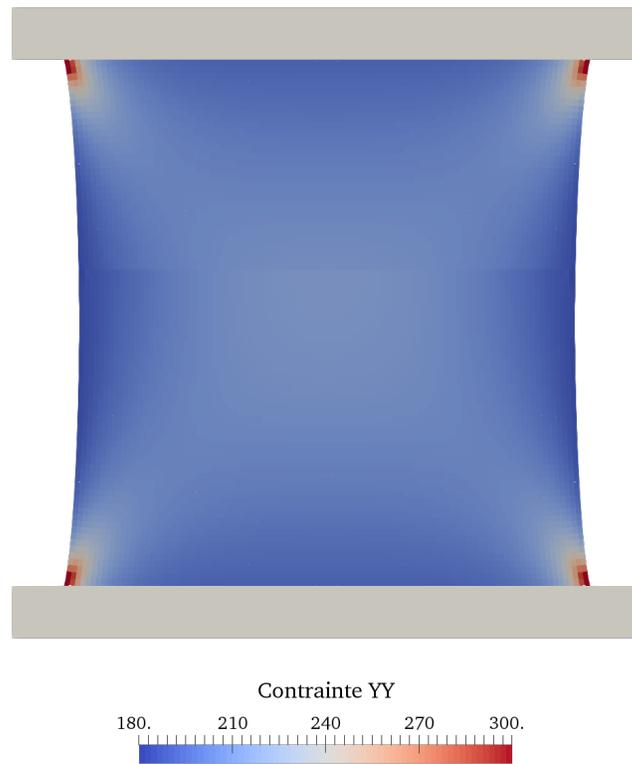
En reprenant la configuration précédemment décrite, nous illustrons l'impact de la pré-charge sur le comportement de la structure. La hauteur et la largeur de la structure globale est L . Un maillage en éléments quadrangles linéaires de côté h est utilisé. Les valeurs des paramètres sont précisées à la table 4.3.

La figure 4.14 présente le champ de contrainte σ_{YY} obtenu après 15 itérations ($\eta \approx 10^{-8}$). Nous avons réussi à imposer une interpénétration conduisant à un champ de contrainte de tension dans la structure. Puisque les deux sous-structures sont constituées du même matériau et que la structure globale est symétrique, chaque sous-domaine s'étire équitablement de la valeur $\mathbf{p}/2$.

Le comportement de pré-charge est sensible à la rigidité des sous-domaines voisins à l'interface de pré-charge. Pour illustrer très simplement cela, nous présentons une comparaison pour un cas où le module de Young E_1 du sous-domaine S_1 est de 50GPa et un second cas pour un module

Paramètres	Valeur
E	200 GPa
ν	0.3
L	10 mm
h	0.1 mm
\mathbf{p}	0.01 mm

TABLE 4.3 – Paramètres

FIGURE 4.14 – Contrainte σ_{YY}

de Young $E_1 = 100\text{GPa}$. Pour ces deux études le module de Young E_2 du second sous-domaine reste fixé à 200GPa . Nous notons respectivement W_1 et W_2 les déplacements normaux moyens des sous-domaines S_1 et S_2 à l'interface.

Du fait que $E_1 \neq E_2$ la symétrie du problème est perdue et ainsi $W_1 \neq -W_2$. Il est cependant vérifié $W_1 - W_2 = p$. Plus un sous-domaine est souple par rapport à l'autre plus il a tendance à se déformer ($|W_2| < |W_1|$). Cela se traduit par des contraintes de tension plus basses lorsque la rigidité globale de la structure est plus faible. Les contraintes σ_{YY} moyennes sont présentées à la table 4.4.

$E_1 = E_2 = 200\text{GPa}$	$E_1 = 100\text{GPa}$	$E_1 = 50\text{GPa}$
235 MPa	157 MPa	95 MPa

TABLE 4.4 – Contraintes σ_{YY} moyenne

Paramétrer une pré-charge est complexe. Ce problème est hyperstatique et les contraintes dépendent donc de la rigidité de la structure. En particulier, en imposant une interpénétration entre deux sous-domaines d'une tige de vis, l'effort global de tension dépend de la rigidité des pièces voisines. Un premier calcul incluant seulement les conditions de pré-charge est souvent réalisé pour contrôler l'état de contrainte. En fonction de la valeur obtenue, l'interpénétration est ensuite ajustée par proportionnalité pour obtenir l'état de contrainte désiré.

4 Application à un cas 3D

Nous nous intéressons à un cas 3D plus complexe faisant intervenir plusieurs interfaces de pré-charge et des interfaces de contact frottant. Deux plaques sont maintenues en contact par l'intermédiaire de trois boulons (Figure 4.15). La symétrie est exploitée pour réduire la taille du problème et conduit à un blocage du déplacement normal (selon z) de la face inférieure de la structure et le déplacement selon y du plan de symétrie (x,z). Les interfaces de contact unilatéral avec jeu correspondent aux interactions entre les tiges des boulons et les plaques. La plaque supérieure décomposée en trois sous-domaines est en contact frottant avec la plaque inférieure elle-même décomposée en trois. Chaque boulon est sous-structuré en trois sous-domaines. Deux parties symétriques correspondent à la tige, un sous-domaine représente l'écrou. Au total le problème comporte 15 sous-structures et 27 interfaces. La description des différentes interfaces et conditions limites est faite à la figure 4.15. Les différents paramètres sont précisés à la table 4.16.

Quatre pas de chargement sont considérés. Le premier correspond à la pré-charge des trois boulons. L'effort F est ensuite appliqué linéairement jusqu'à atteindre la valeur maximale de 30MPa.

La figure 4.18a montre les éléments en compression quand la pré-charge est appliquée. On retrouve le classique cône de Rotscher sous les têtes de vis [Rotscher, 1927]. Les tiges sont soumises à un état de contrainte de tension (Figure 4.18b). La figure 4.19 montre la carte de contrainte de Von Mises, la déformée de la structure est amplifiée par 500. Nous observons un glissement entre les deux plaques ce qui conduit à un contact entre les tiges des boulons et les plaques. La répartition de la contrainte de Von Mises montre une structure sollicitée de toute part. Des efforts tangentiels sont bien transmis grâce au frottement.

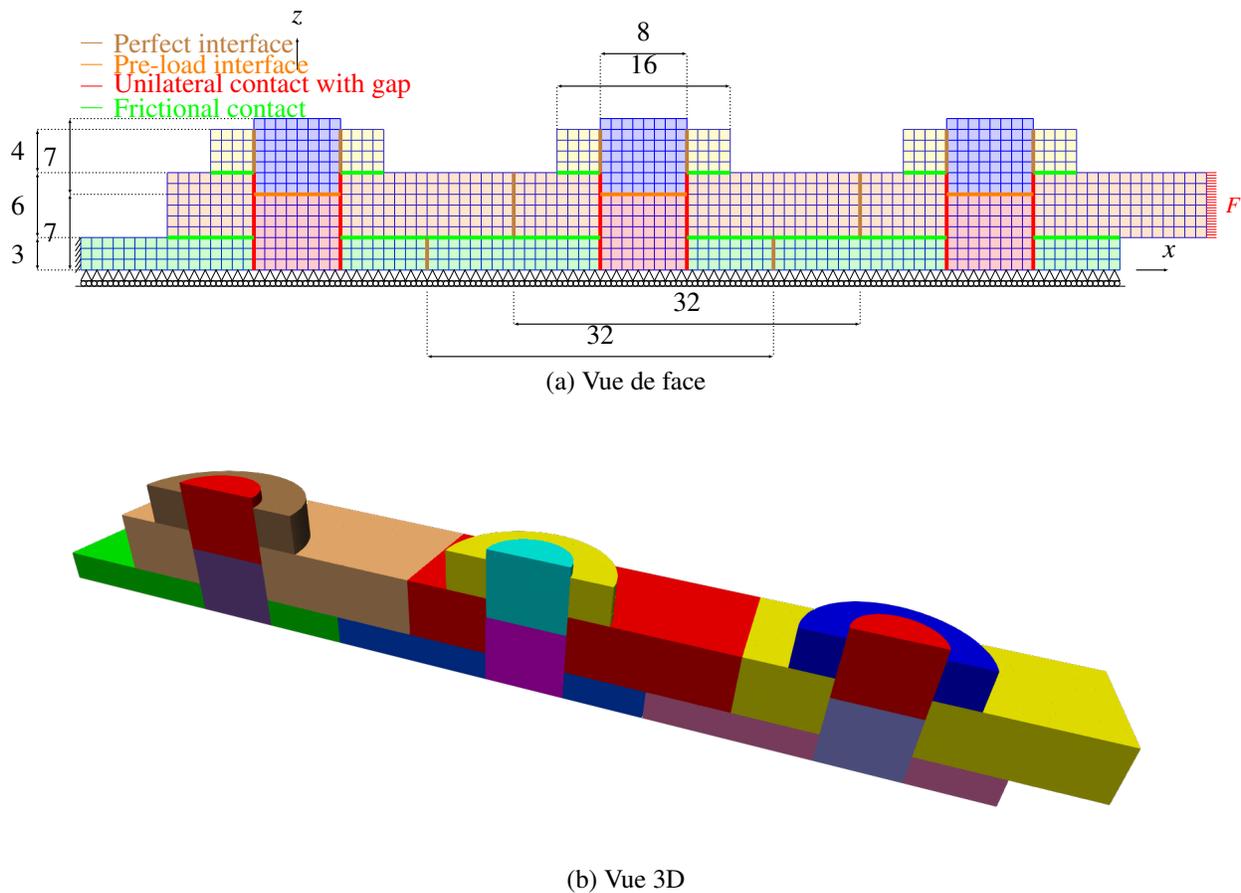


FIGURE 4.15 – Description de l'assemblage 3D

5 Bilan

Nous nous sommes intéressés dans ce chapitre à l'étape locale de la méthode Latin. En particulier après avoir introduit la formulation quasi-statique pour l'étape linéaire, ainsi que l'étape locale pour des interfaces parfaites, nous avons présenté la résolution explicite du problème de contact frottant avec des notations algébriques.

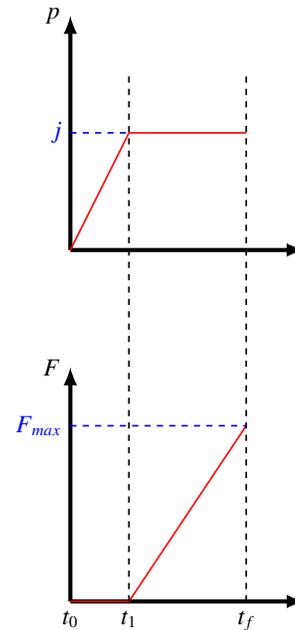
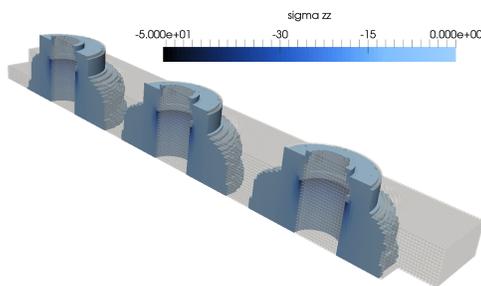
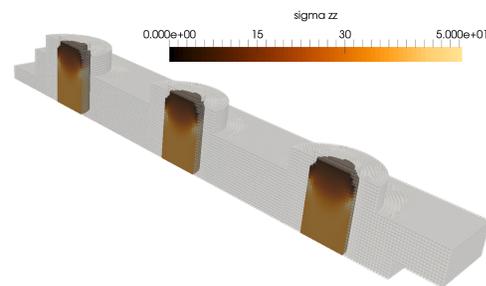
Afin de simplifier la résolution du problème et conserver le caractère local du comportement de contact, il est nécessaire de modifier la direction de recherche non-intrusive. \mathbf{k}^+ couple actuellement les noeuds d'interface entre eux. En choisissant d'extraire la diagonale de l'opérateur de rigidité des semelles, la direction de recherche redevient locale et permet de résoudre explicitement le problème de contact frottant.

Ce problème est résolu en deux étapes. Un premier indicateur est déterminé afin de prédire le statut des noeuds en contact (*contact* ou *décollement*) pour ensuite calculer les composantes normales. Un second indicateur est évalué pour les noeuds en contact prédisant le statut d'adhérence ou de glissement des noeuds.

Nous avons défini des interfaces de type pré-charge en imposant des contraintes spécifiques aux composantes normales des champs d'interface. En particulier une interpénétration est mise en

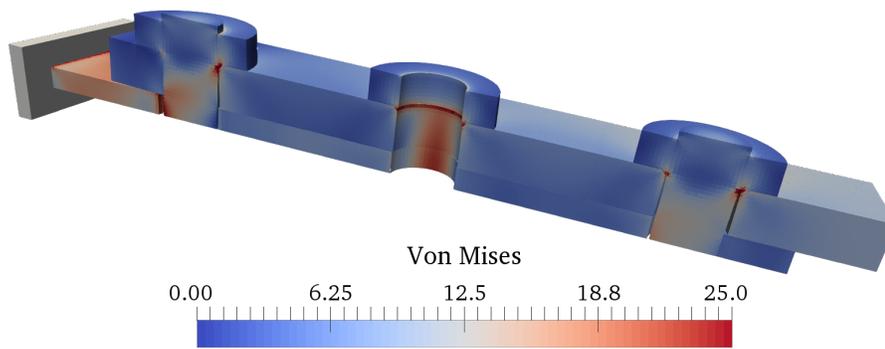
Paramètres	Valeur
E	200 GPa
ν	0.3
pré-charge j	0.001mm
pas de temps Δt	1s
Nombre de pas	4
F_{max}	30 MPa
μ	0.1
jeu j_n	0.001mm
taille de maille h	0.5mm

FIGURE 4.16 – Description des interfaces

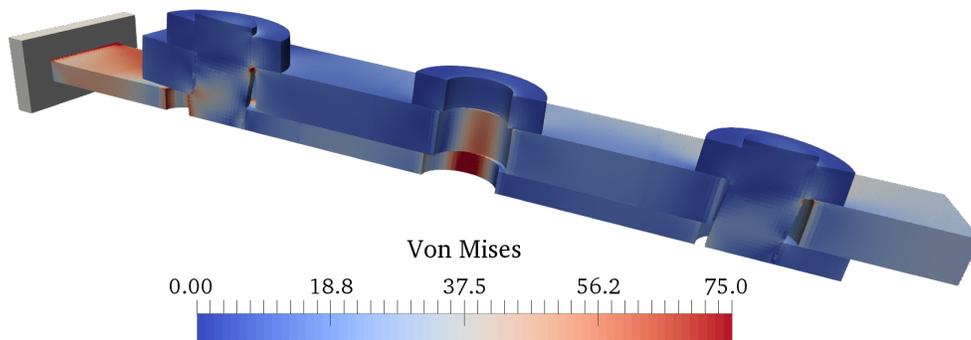
FIGURE 4.17 – Histoire du chargement F (a) Cônes de Rotscher dus aux pré-charges : $\sigma_{zz} \leq 0$ (b) Tension dans les tiges : $\sigma_{zz} \geq 0$ FIGURE 4.18 – Contrainte en MPa au premier pas de chargement t_1

place afin d'assurer un état de contrainte de tension dans les éléments boulonnés. La relation de non-interpénétration avec jeu initial est alors inversée. Les conditions normales portant sur l'indicateur de contact s'inversent donc aussi, et conduisent à déterminer une solution de type décollement pour $C_n < 0$ et de contact pour $C_n \geq 0$. Les composantes tangentielles sont déterminées de telle sorte qu'une continuité du déplacement tangentiel et un équilibre des efforts tangentiels soient assurés.

Nous avons désormais présenté les différents ingrédients permettant de traiter des problèmes complexes. Nous illustrerons donc dans le chapitre suivant quelques applications à des assemblages industriels.



(a) Contrainte de Von Mises - déformée $\times 500 - t_2$



(b) Contrainte de Von Mises - déformée $\times 500 - t_4$

FIGURE 4.19 – Contrainte de Von Mises

Implémentation numérique et exemples industriels

Nous abordons dans ce chapitre des éléments ayant trait à l'implémentation de la méthode Latin au sein de code_aster. Une première section présente les principes généraux de code_aster. Une deuxième section détaille plus précisément les outils développés au sein de ces travaux de thèse. Une troisième section illustre les performances de l'implémentation non-intrusive sur des cas tests académiques. Une dernière section présente des simulations sur des structures industrielles.

Sommaire

1	Introduction - principes et structures de <i>code_aster</i>	133
1.1	Principes de <i>code_aster</i> et du superviseur	133
1.2	Base de données JEVEUX - concept ASTER	134
2	Détails de l'implémentation et mise en donnée	135
2.1	Pré-traitement	135
2.1.1	Mise en donnée xml	135
2.1.2	Sous-structuration et construction du maillage	136
2.1.3	Pondération automatique de la direction de recherche	137
2.2	Calcul Latin générique	138

2.2.1	Plusieurs niveaux de parallélisme	138
2.2.2	Gestion de la mémoire	139
3	Performance de la méthode pour des cas académiques	141
3.1	Etude du <i>speed-up</i>	141
3.2	Cas de contacts multiples	142
3.2.1	Comparaison des deux études	142
3.2.2	Quelques résultats informatiques	143
4	Quelques applications industrielles	148
4.1	Liaison de lanceur spatial	148
4.2	Cas test industriel	150
4.2.1	Sous-structuration initiale	150
4.2.2	Re-sous-structuration pour équilibrer les sous-domaines	153
5	Bilan	156

1 Introduction - principes et structures de *code_aster*

L'objectif de cette première section est de présenter les principes généraux de *code_aster*, ainsi que les spécificités de structures de données pour comprendre certains aspects de l'implémentation non-intrusive.

1.1 Principes de *code_aster* et du superviseur

Pour réaliser une étude avec *code_aster*, il est nécessaire de fournir un maillage ainsi qu'un fichier de commande. Le maillage est généralement sous format *MED* et généré à l'aide de *Salome*. Le fichier de commande est écrit sous une syntaxe python et contient des fonctions particulières dont le langage est spécifique à *code_aster*. Ainsi l'utilisateur décrit son fichier de commande dans un environnement python et peut bénéficier des nombreuses fonctionnalités disponibles : bibliothèques de calculs scientifiques, *profiling* du code...

Les fonctions appelées dans le script de commandes sont généralement décrites dans les sources de *code_aster* en fortran. Cet environnement fortran n'est pas accessible à l'utilisateur. Ainsi on pourrait résumer la structure de *code_aster* à deux environnements : l'environnement utilisateur en python et l'environnement d'exécution ASTER en fortran (espace JEVEUX). Afin de faire dialoguer ces deux environnements il existe une couche intermédiaire appelée *superviseur* (Figure 5.1).

Par exemple lorsqu'une fonction est appelée dans le script de commande, le superviseur va activer l'opérateur dans l'environnement fortran. En fonction des arguments fournis à l'opérateur, une structure de données appelée *concept* est créée dans l'environnement fortran. Le superviseur se charge ensuite de faire le lien entre ce concept situé dans le fortran et le concept qui est ensuite renvoyé dans l'environnement python.

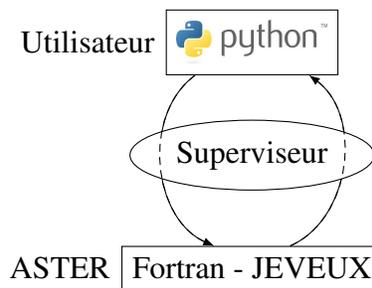


FIGURE 5.1 – Principe du superviseur

1.2 Base de données JEVEUX - concept ASTER

L'ensemble des objets propres à *code_aster* est stocké dans des bases de données dans l'environnement fortran. Deux bases sont créées lors de l'exécution d'une étude ASTER :

- une base VOLATILE contenant l'ensemble des objets de travail temporaires et intermédiaires des différents opérateurs (hors superviseur). Elle est aussi vidée à la fin de l'exécution du code.
- une base GLOBALE regroupant l'ensemble des concepts relatifs aux fonctions exécutées dans le script de commande python. Cette base utilisateur est sauvegardée à la fin de l'exécution du code.

Les concepts utilisateurs, qui sont les objets renvoyés par les fonctions du script de commandes python, sont des structures de données regroupant différents objets JEVEUX élémentaires. Chaque objet JEVEUX élémentaire contient un ensemble d'informations homogènes comme des entiers, réels, chaînes de caractères... Ces concepts sont stockés dans la base globale et le superviseur se charge de faire le lien entre cette base de données et l'environnement python. La couche intermédiaire du superviseur empêche toute manipulation directe des attributs des concepts. En fait il n'est retourné dans l'environnement python que le type de structure de données ainsi que l'adresse mémoire du concept dans la base globale.

Dans l'exemple ci-dessous, un concept matériau nommé *mater* est créé. On définit un matériau élastique linéaire isotrope de module de Young de 200 GPa, de masse volumique 7000 kg.m⁻³ et de coefficient de Poisson de 0.3. Lorsque que l'on demande d'afficher *mater*, il est seulement retourné son type (structure de données ASTER de type matériau : *mater_sdaster*) ainsi que son adresse dans la base (0x7fa89b996590).

```

1 mater = DEF_MATERIAU(ELAS=_F(E=200000 ,
2                               RHO=7000 ,
3                               NU=0.3 ,),
4                               )
5 >>> mater
6 <Cata.cata.mater_sdaster object at 0x7fa89b996590>
```

Remarque 5.0

Si l'on souhaite tout de même avoir accès aux caractéristiques matériaux depuis l'environnement python, il est possible d'utiliser des méthodes lourdes et spécifiques :

```
1 >>> aster.getvectjev("mater" .CPT.0 00001 .VALR")
2 (200000.0 , 7000.0 , 0.3)
```

Cela nécessite cependant de connaître précisément la structure de données JEVEUX. L'implémentation non-intrusive doit alors éviter d'avoir recours à ces appels aux informations directement dans l'environnement fortran.

Lorsqu'un concept est créé via une fonction dans le script de commande un espace mémoire lui est attribué au sein de la base JEVEUX. Le nom du concept ne peut pas être directement re-utilisé. Par exemple si l'on souhaite changer le matériau mater en le remplaçant directement par un autre matériau comme suit :

```
1 mater = DEFI_MATERIAU(ELAS = _F(E = 100000 ,
2                               NU = 0.3 ,
3                               RHO = 7000 ,
4                               )
5 )
```

un erreur se produit car le nom de concept est déjà utilisé :

```
1 ERREUR A L'INTERPRETATION DANS ACCAS – INTERRUPTION
2 >> JDC.py : DEBUT RAPPORT
3 CR phase d'initialisation
4 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
5 ! Etape DEFI_MATERIAU ligne : 22 fichier : fort.1 Nom de concept deja defini : !
6 ! mater !
7 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
8 fin CR phase d'initialisation
```

Il est alors nécessaire de détruire au préalable le concept mater avec la fonction DETRUIRE :

```
1 DETRUIRE(CONCEPT=_F(NOM=mater))
```

Le nom de concept peut ensuite être utilisé sans problème.

2 Détails de l'implémentation et mise en donnée

2.1 Pré-traitement

2.1.1 Mise en donnée xml

La mise en donnée se fait par l'intermédiaire d'un fichier xml permettant une structuration efficace des données. Ce fichier regroupe l'ensemble des paramètres propres à la méthode : le critère d'erreur voulue, le choix d'une approche multi-échelle, l'hypothèse quasi-statique..., mais aussi les informations relatives à la structure étudiée comme : le nombre de sous-structures, les caractéristiques matériaux, les conditions limites, la définition des interfaces... Ce format de mise en donnée

a pour avantage d'être lisible aisément par python en sélectionnant directement les informations qui nous intéressent au sein du fichier.

Par exemple, la définition des conditions limites de Dirichlet s'effectue simplement au LISTING 5.1 en déclarant le nombre total de conditions de Dirichlet en attribut de la balise `dirichlet`. Chaque condition limite est ensuite décrite indépendamment par les balises `dirichletCond` et sont référencées via leur attribut `id`. Les différentes composantes impactées sont décrites par la balise `component` dont l'attribut `spécifie` la composante et dont le contenu précise la valeur de la condition. Ainsi cet exemple représente une condition limite d'encastrement. Un blocage du seul déplacement selon x aurait consisté à n'écrire qu'une seule balise `component`. Ce formalisme permet d'être extrêmement générique et souple dans la définition des données.

```

1 <dirichlet N="1">
2   <!-- 1st Dirichlet condition -->
3   <dirichletCond id="0">
4     <component comp="DX">0.</component>
5     <component comp="DY">0.</component>
6     <component comp="DZ">0.</component>
7   </dirichletCond>
8 </dirichlet>

```

Listing 5.1 – Exemple de définition de conditions limites

2.1.2 Sous-structuration et construction du maillage

Les seules conditions que doit respecter le maillage de la structure étudiée sont de contenir les groupes de mailles relatives aux conditions limites : ils sont nommés \bar{w}_i , \bar{f}_j , respectivement le groupe relatif à la i -ème condition de Dirichlet et le groupe relatif à la j -ème condition limite de Neumann. Le maillage doit aussi contenir les groupes de mailles nommés S_k relatifs à chaque sous-domaines S_k . La phase de pré-traitement consiste à construire à partir de ce premier maillage l'ensemble des interfaces et connectivités entre les sous-domaines, les semelles associées, les groupes de conditions limites pour chaque sous-domaine... Ce pré-traitement est scripté dans *Salome-meca* et permet de traiter de manière complètement générique n'importe quelle structure. Pour résumer cette phase, deux étapes sont importantes : la détection et création des interfaces, et la création des semelles.

Création des interfaces : Une première étape consiste à créer les mailles de bord $(b_k)_k$ des sous-domaines $(S_k)_k$. Cela permet ensuite de tester pour chaque sous-domaine ceux qui ont des mailles de bords en commun, définissant ainsi une interface (Algorithme 2.1)

Algorithme 2.1 : Création des interfaces

```

for  $S_k \in \mathcal{E}$  do
  for  $S_j \in \mathcal{E} \setminus k$  do
    if  $b_k \cap b_j \neq \emptyset$  then Mailles en commun
      └ Création d'une interface :  $\Gamma_{kj} = b_k \cap b_j$ 

```

 **Remarque 5.1**

Cette méthode n'est utilisable que dans le cas de maillages compatibles.

Création des semelles Une fois les interfaces détectées et les groupes de mailles associés créés, nous construisons les semelles relatives à ces interfaces comme étant les éléments des voisins dont au moins un noeud appartient aux interfaces. Concrètement en considérant l'interface Γ_{kj} , la semelle θ_{kj} du sous-domaine S_k est constituée des éléments du sous-domaine S_j ayant au moins un noeud sur l'interface Γ_{kj} . La figure 5.2 présente un exemple de semelles.

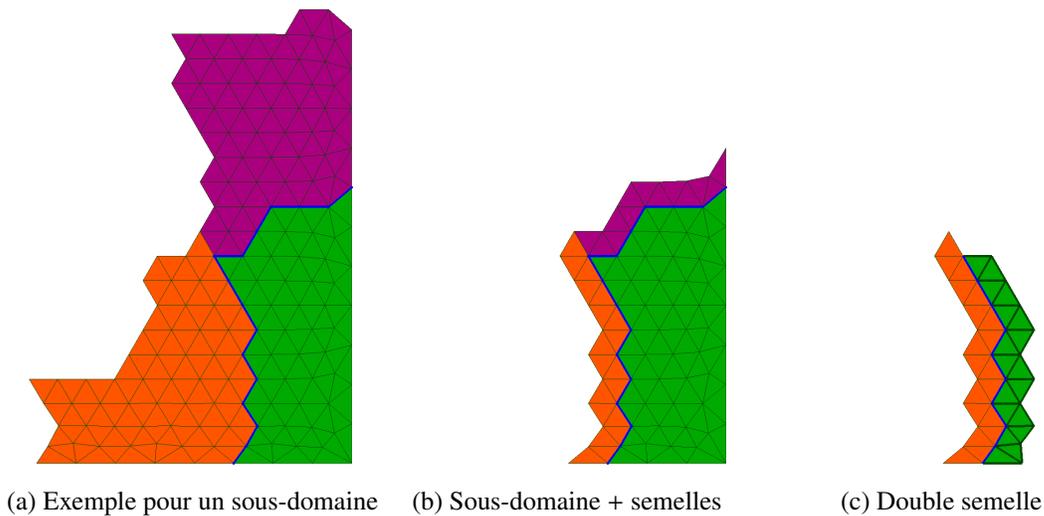


FIGURE 5.2 – Exemple de semelles

2.1.3 Pondération automatique de la direction de recherche

Nous avons vu que la direction de recherche doit avoir une rigidité comparable à la rigidité de la structure globale. Cependant cette notion de rigidité dépend des dimensions du problème et les pondérations ne sont pas nécessairement les mêmes pour toutes les directions. Une "optimisation" pour chaque interface dépend de la rigidité globale de la structure, ainsi que des sous-domaines voisins. Ce procédé lourd ne peut être aisément implémenté de façon non-intrusive. Ainsi pour obtenir un caractère générique, un compromis doit être fait, et ces pondérations "optimales" ne sont pas déterminées. La pondération des caractéristiques matériaux de la semelle est choisie de telle sorte que la rigidité de la semelle suivant la direction normale moyenne à l'interface soit égale à la rigidité du sous-domaine associé suivant la même direction.

Plus concrètement on détermine à l'aide de *Salome-Meca* les boîtes englobantes (*bounding boxes*) des sous-domaines ainsi que des semelles (exemple Figure 5.3). Une normale à chaque élément de l'interface est déterminée facilement sous *code_aster* (opérateur `CREA_CHAMP` avec l'option `OPERATION='NORMALE'`). Cela nous permet d'obtenir une normale moyenne à l'interface.

En notant $\Delta SD = [\Delta X, \Delta Y, \Delta Z]$ la boîte englobante du sous-domaine, $\Delta s = [\Delta x, \Delta y, \Delta z]$ celle de la semelle et \underline{n} la normale moyenne à l'interface, le module de Young E_s de la semelle est choisi en fonction du module de Young E_{SD} du sous-domaine tel que :

$$\frac{E_s}{|\Delta s \cdot \underline{n}|} = \frac{E_{SD}}{|\Delta SD \cdot \underline{n}|} \quad (5.1)$$

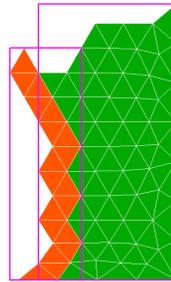


FIGURE 5.3 – Exemple de *bounding box*

2.2 Calcul Latin générique

2.2.1 Plusieurs niveaux de parallélisme

On peut évoquer plusieurs niveaux de parallélismes au cours de la résolution du problème sous-structuré. Un premier niveau fait appel au protocole MPI et se situe dans l'environnement python. On lance pour chaque sous-domaine une instance de *code_aster*. Ces instances effectuent en parallèle le même algorithme. Elles échangent entre elles des objets python grâce au paquet `mpi4py` de python. Un autre niveau de parallélisme peut être envisagé pour la résolution des problèmes linéaires sur les sous-domaines. Un appel à une résolution *multi-thread* par le solveur *MUMPS* inclus dans *code_aster* est possible. Nous n'utilisons cependant pas ce *multi-thread* de *code_aster* et restons mono-coeur par sous-domaine. Ce principe général est décrit à la figure 5.4.

Pour chaque instance de *code_aster* :

- à l'étape linéaire, un problème est résolu seulement sur le sous-domaine associé au processus *MPI*,

Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages

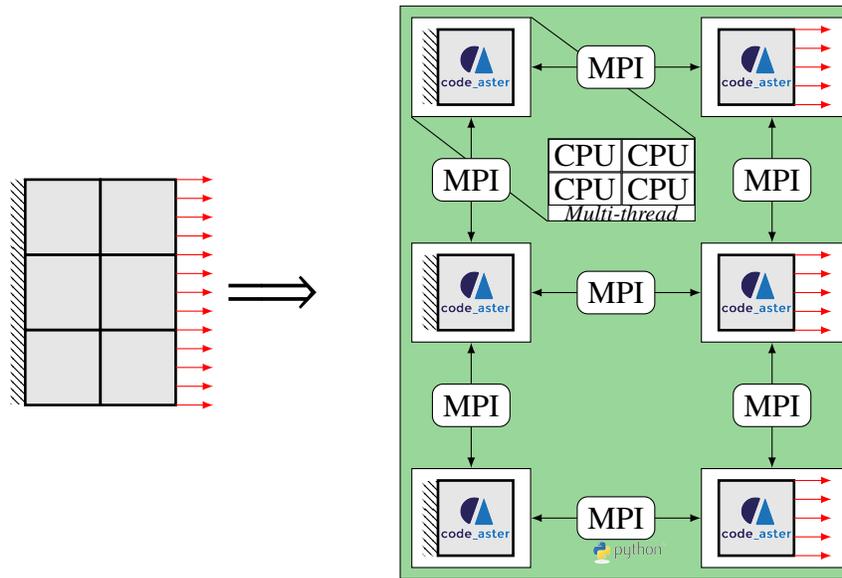


FIGURE 5.4 – Principe de l'implémentation parallèle

- à l'étape locale, sont résolus les problèmes relatifs aux interfaces associées au sous-domaine en question.

Remarque 5.2

Les problèmes d'interface de l'étape locale sont dupliqués. Pour l'interface Γ_{ij} une résolution est effectuée par le processus *MPI* relatif au sous-domaine S_i , et une seconde est effectuée en parallèle par le processus *MPI* relatif au sous-domaine S_j . Les sous-domaines voisins effectuent donc en parallèle le même problème d'interface. Cela permet d'éviter une communication entre les processus *MPI* à la fin de l'étape locale. Ainsi un processus *MPI* traite non seulement un sous-domaine, mais aussi l'ensemble des interfaces le concernant.

2.2.2 Gestion de la mémoire

Nous avons précisé à la section 1.2 que pour réutiliser des noms de concept déjà définis il est nécessaire de les détruire en appelant l'opérateur *DETRUIRE*. Cet opérateur ne détruit pas seulement le pointeur vers l'adresse mémoire dans la base *JEVEUX* mais aussi le contenu à l'adresse indiquée.

Pour comprendre ces problématiques de nommage et de mémoire nous utilisons un exemple très simple. Nous décrivons (en simplifiant) la fonction qui permet de résoudre un problème à déplacement imposé sur une semelle. En argument sont donnés la classe relative à la semelle contenant le modèle, le matériau et les conditions limites d'encastrement; ainsi que le champ de déplacement à imposer. Une première étape consiste à affecter le champ en concept de conditions limites via *AFFE_CHAR_CINE*. Ensuite la résolution du problème est effectuée via *MECA_STATIQUE*.

```

1 def resoudreSemelle(semelle, champ):
2     ### Affectation du champ en conditions limites
3     CL = AFFE_CHAR_CINE(MODELE=semelle.modele.MODELE,
4                         MECA_IMPO=_F(GROUP_MA=semelle.maillage.GNO['CL'],
5                                     DX=champ['DX'],DY=champ['DY'],
6                                     )
7                         )
8     ### Resolution
9     res = MECA_STATIQUE(MODELE=semelle.modele.MODELE,
10                        CHAM_MATER=semelle.modele.MATERIAU,
11                        EXCIT=(_F(CHARGE=semelle.CL),
12                               _F(CHARGE=CL)
13                        )
14                        )
15     return res

```

Le problème avec cette écriture est le fait que lorsque cette fonction sera de nouveau appelée, le concept CL aura déjà été défini une première fois : les concepts ASTER sont globaux durant l'exécution du code. Il est donc nécessaire à la fin de la fonction de "nettoyer" les grandeurs qui ne sont pas nécessaires pour la suite de l'exécution. Un appel à la fonction DETRUIRE est inévitable.

Puisque `res` est la sortie de la fonction, il n'est pas possible de la détruire. Pour cela nous devons sauvegarder les concepts nécessaires au sein d'une liste python que nous incrémentons à chaque fois qu'un concept doit être sauvegardé. Le pointeur créé en sortie de la fonction est alors directement dirigé vers l'élément de la liste qui est déclarée globale. Au début du fichier contenant la fonction, sont donc créés une grande liste `l` d'éléments vides ainsi qu'un itérateur python `it`. Avec ces notations la fonction devient :

```

1 dim=1000000
2 l=[None]*dim
3 it=iter(range(dim))
4
5 def resoudreSemelle(semelle, champ):
6     global l, global it
7     ### Affectation du champ en conditions limites
8     CL = AFFE_CHAR_CINE(MODELE=semelle.modele.MODELE,
9                         MECA_IMPO=_F(GROUP_MA=semelle.maillage.GNO['CL'],
10                                     DX=champ['DX'],DY=champ['DY'],
11                                     )
12                         )
13     ### Resolution
14     kk=it.next()
15     l[kk] = MECA_STATIQUE(MODELE=semelle.modele.MODELE,
16                          CHAM_MATER=semelle.modele.MATERIAU,
17                          EXCIT=(_F(CHARGE=semelle.CL),
18                                  _F(CHARGE=CL)
19                          )
20                          )
21     DETRUIRE(CONCEPT=_F(NOM=CL))
22     return l[kk]

```

Ainsi le concept de conditions limites CL, qui varie à chaque appel de la fonction, est détruit à la fin. Le pointeur du résultat de l'opérateur MECA_STATIQUE est enregistré dans la liste globale `l` pour être réutilisé ensuite dans le code.

Cette obligation de sauvegarder les concepts au cours de l'exécution du code devient très handicapante d'un point de vue mémoire. La liste servant à stocker les différents concepts grossit constamment et des limitations en mémoire peuvent apparaître. Certains concepts n'ont pas besoin d'être conservés tout au long de l'exécution. Les résultats des problèmes mécaniques sur les sous-domaines et sur les semelles varient à chaque itérations. Par exemple, ceux déterminés à la première itération peuvent être supprimés à la fin de la deuxième itération. Avec une telle procédure de nettoyage de la base globale on limite fortement la surcharge en mémoire.

3 Performance de la méthode pour des cas académiques

3.1 Etude du *speed-up*

Nous proposons ici une étude de l'influence de la parallélisation sur le temps de résolution du problème sous-structuré (ou aussi étude du *speed-up*). Nous reprenons la structure poutre 3D de la section 4 du chapitre 3 mais cette fois-ci maillée volontairement plus finement. Alors que pour illustrer l'impact de l'approche multi-échelle la structure possède environ 70000 degrés de liberté, ici la structure possède près de 25 millions de degrés de liberté. Le critère d'erreur choisi est de 10^{-5} . Cette étude de *speed-up* a été réalisée grâce au mesocentre *fusion* de CentraleSupélec et de l'ENS Paris-Saclay. Le gain en temps CPU est normalisé par rapport au temps CPU pour la sous-structuration à 16 sous-domaines.

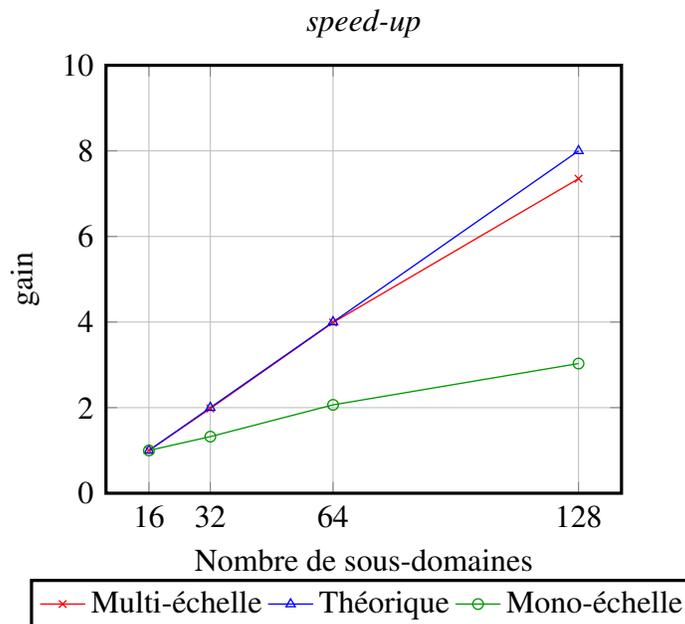


FIGURE 5.5 – *Speed-up*

La figure 5.5 permet de comparer l'accélération obtenue par rapport à une accélération théorique. Plus nous avons de sous-domaines dans l'approche mono-échelle, plus ce *speed-up* se dégrade. En effet, bien que le temps de calcul des itérations diminue en augmentant le nombre de

sous-domaines, il est nécessaire d'effectuer plus d'itérations pour atteindre un niveau de convergence similaire que pour un cas avec moins de sous-domaines. L'extensibilité de l'approche multi-échelle permet de palier ce problème. Nous avons vu à la section 4 du chapitre 3 que le nombre d'itérations était équivalent quelle que soit la sous-structuration. Ainsi pour une approche multi-échelle le *speed-up* est fortement amélioré et tend vers le *speed-up* théorique optimal.

Remarque 5.3

Le *speed-up* pour plusieurs milliers de sous-domaine ne tendrait pas vers le *speed-up* optimal. Plus le nombre de sous-domaines augmentent plus les temps de communication deviennent prépondérants.

3.2 Cas de contacts multiples

Afin d'illustrer le comportement de la méthode Latin en présence de nombreuses interfaces de contact, nous choisissons de traiter une structure de type polycristalline similaire à celle de la section 5. Le cas test proposé ici comporte 100 sous-domaines, ainsi qu'environ 565 000 degrés de liberté. Au total la structure comporte 467 interfaces qui sont toutes définies comme des interfaces de type contact frottant. Le bord inférieur est encastré, les bords latéraux sont libres et un effort normal est appliqué sur le bord supérieur dont l'évolution est précisée à la figure 5.7. Deux cas correspondant à des coefficients de frottement de 0.1 et 0.6 sont traités.

3.2.1 Comparaison des deux études

La figure 5.8 montre la déformée de la structure au cours des pas de chargement pour un coefficient de frottement de 0.1. De nombreux glissements et décollements apparaissent pendant la charge de la structure polycristalline. Lors de la décharge, à cause du faible coefficient de frottement et du décollement, la structure ne peut revenir dans sa configuration initiale et du jeu persiste aux interfaces. La figure 5.9 illustre quant à elle la déformée pour un coefficient de frottement de 0.6. Ce coefficient élevé conduit à des interfaces fortement adhérentes. Les sous-domaines ne glissent que très peu les uns par rapport aux autres et très peu de jeu apparaît aux interfaces. Pendant la charge la structure reste compacte. Lors de la décharge, même si de légers décollements persistent, la solution revient quasiment à l'état initial.

Nous comparons à la figure 5.10 la convergence des deux études. Le multi-échelle apporte un gain de convergence au début des itérations. La convergence multi-échelle est similaire pour les deux études sur les premières itérations. Cependant son impact s'estompe plus rapidement lorsqu'il y a un glissement important aux interfaces. Dans le cas de faibles coefficients de frottement, de nombreuses interfaces sont en situation de décollement, et aucun effort n'est transmis à ces interfaces. Le multi-échelle ne peut donc pas apporter d'informations supplémentaires. Les solutions d'adhérence et de glissement sont plus complexes à déterminer que les solutions de type décollement expliquant que la convergence pour des coefficients de frottement élevés est moins bonne que celle pour des coefficients de frottement faibles.

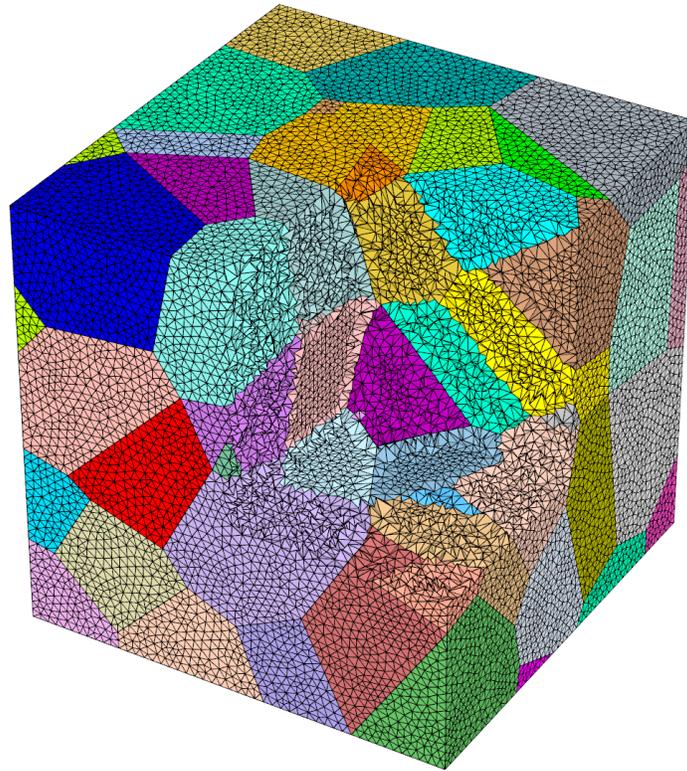


FIGURE 5.6 – Sous-structuration et maillage

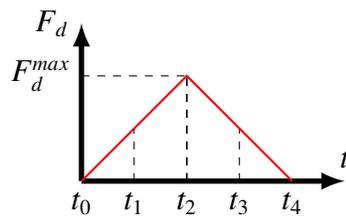


FIGURE 5.7 – Description du chargement : 4 pas de charge

3.2.2 Quelques résultats informatiques

Nous présentons ici quelques données relatives à l'outil développé pour piloter *code_aster*. L'objectif est de montrer que l'utilisation faite de *code_aster* est intensive et inhabituelle. Pour cela nous avons effectué un *profile* python de l'exécution. En particulier nous nous sommes intéressés seulement à la phase itérative et avons mis de côté la phase de préparation et de factorisation

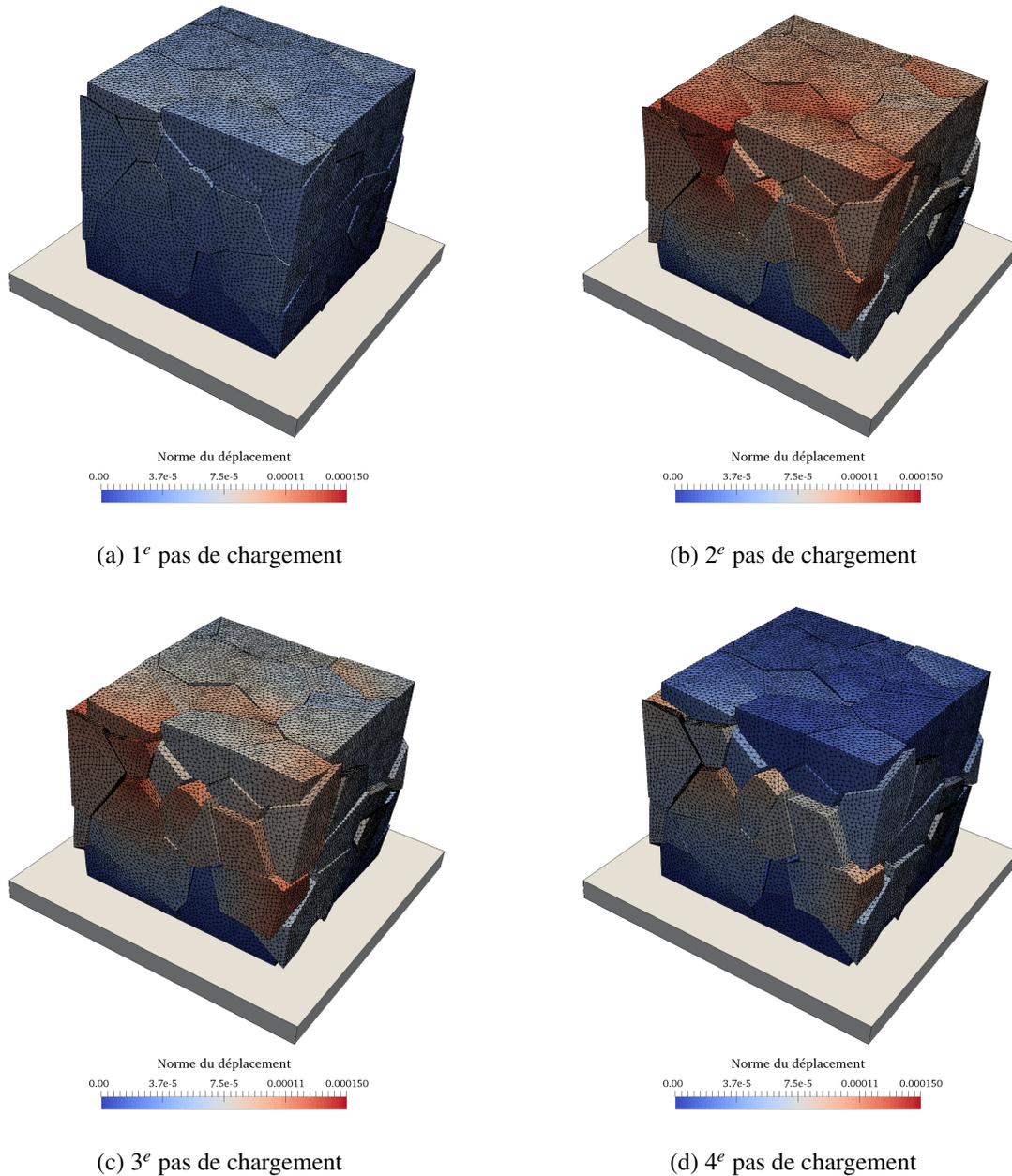


FIGURE 5.8 – Déformée à chaque pas de chargement ($\times 1000$) pour un coefficient de frottement de 0.1

des opérateurs. Nous présentons le nombre d'appels et le temps d'exécution cumulé de certaines fonctions. Nous indiquons aussi la part que cela représente par rapport au temps d'exécution total. Certaines des fonctions listées sont appelées par d'autres. Il y a donc "recouvrement" des parts du temps d'exécution que nous présentons en %.

Nous avons ciblé le plus petit et le plus gros des sous-domaines pour comparer les différents

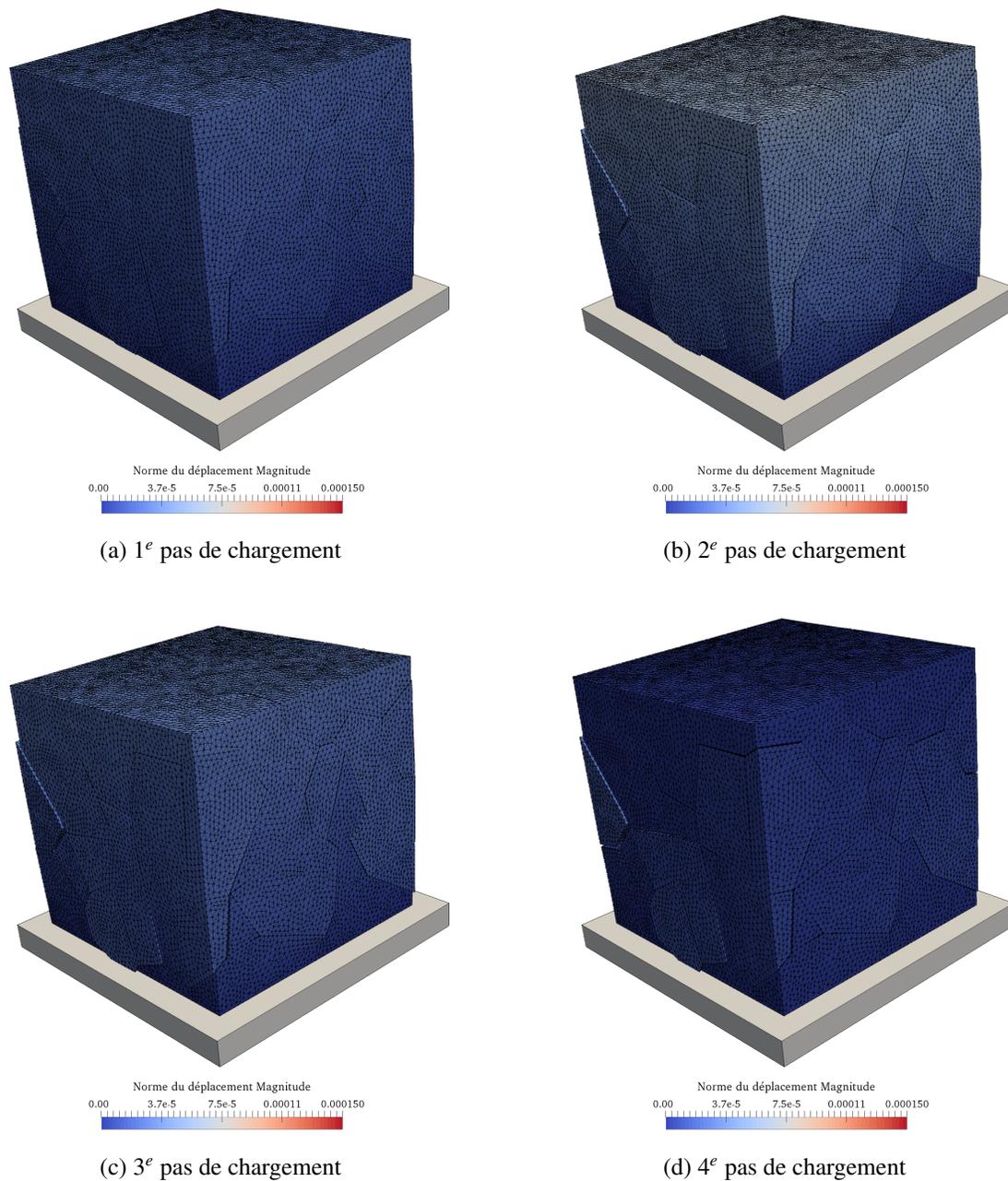


FIGURE 5.9 – Déformée à chaque pas de chargement ($\times 1000$) pour un coefficient de frottement de 0.6

temps de calcul. Les données relatives au nombre d'interfaces ou encore au nombre de noeuds pour ces deux sous-domaines sont précisées à la table 5.1

Les détails des fonctions principales sont données au tableau 5.2. Les parts que représentent chaque fonction sont différentes entre les deux sous-domaines. Le nombre plus important d'inter-

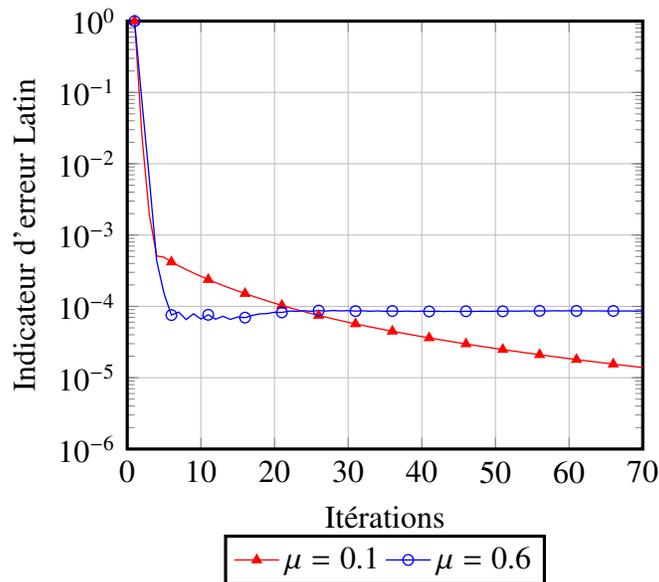


FIGURE 5.10 – Comparaison de la convergence

	Petit sous-domaine	Gros sous-domaine
Noeuds au total	529	5639
Noeuds d'interface	160	1258
Nb interfaces	5	18

TABLE 5.1 – Comparaison entre le plus petit et le plus gros sous-domaines

faces et de noeuds d'interface pour le plus gros sous-domaine par rapport au plus petit implique que l'étape locale est plus de 5 fois plus longue, et représente près de 27% du temps total d'exécution (contre seulement 4.84% pour le plus petit). Puisque le plus petit sous-domaine finit l'étape locale plus rapidement, il enchaîne directement avec l'étape linéaire. Cependant au cours de l'étape linéaire, une communication globale *allreduce* est nécessaire pour l'approche multi-échelle. Le plus petit des sous-domaines doit donc inévitablement attendre le plus gros. Cela explique pourquoi l'étape linéaire est plus longue. Le temps plus long pour communiquer les champs d'interfaces aux sous-domaines voisins s'explique aussi de la même façon. Les quelques pourcentages de la part du temps d'exécution restant correspondent à des fonctions permettant principalement de sauvegarder les résultats et certaines informations d'interface pour le post-traitement.

Le tableau 5.3 présente le détail pour certaines fonctions intervenant dans les fonctions principales décrites au tableau 5.2. En particulier nous présentons les détails concernant la fonction permettant une résolution sur les semelles, la fonction qui résout un problème sur les sous-domaines, la fonction python *sendrecv* et *allreduce* qui servent à communiquer les grandeurs d'interface et échanger les contributions de chaque sous-domaine au calcul de l'indicateur d'erreur et au problème macroscopique. La fonction *DETRUIRE* de *code_aster* est aussi présentée. Avec ces résultats, on montre qu'une part non négligeable du temps de calcul est passée dans la résolution des problèmes sur les semelles. Bien que s'effectuant rapidement (en moyenne 0.6s pour le plus gros

	Petit sous-domaine			Gros sous-domaine		
	Nombre d'appel	Temps d'exécution	Part globale	Nombre d'appel	Temps d'exécution	Part globale
Étape linéaire	70	26341	72.2%	70	15954	43.7%
Étape locale	70	1766	4.84%	70	9964	27.3%
Calcul indicateur	70	5200	14.2%	70	4677	12.8%
Communication des champs	70	1386	3.8%	70	38	0.1%
Nettoyage des bases	70	828	2.3%	70	4944	13.5%
Total			97.3%			97.4%

TABLE 5.2 – Détails du temps d'exécution des fonctions principales

	Sous-domaine 15			Sous-domaine 30		
	Nombre d'appel	Temps d'exécution	Part globale	Nombre d'appel	Temps d'exécution	Part globale
Problème semelles	10500	3845	10.5%	37800	25564	70.1%
Problème sous-domaine	350	540	1.5%	350	2696	7.4%
Python mpi.allreduce	630	28615	78.4%	630	815	2.2%
Python mpi.sendrecv	1750	1385	3.8%	6300	37	0.1%
DETRUIRE	62495	1445	4.0%	215953	8665	23.7%

TABLE 5.3 – Détails du temps d'exécution de certaines autres fonctions

sous-domaine), à cause du grand nombre d'appels à la fonction (37 800), sa part représente 70% du temps total de calcul. La résolution des problèmes sur les sous-domaines n'est finalement pas le plus gros consommateur de temps de calcul. L'analyse des fonctions *sendrecv* et *allreduce* montre que le plus petit sous-domaine occupe plus de 80% de son temps à attendre pour échanger des données alors que le plus gros des sous-domaines n'y consacre même pas 3%. La fonction DETRUIRE, qui est présente dans le nettoyage des bases mais aussi dans chaque fonction qui fait appel à des opérateurs Aster, est utilisée plus de 200 000 fois pour le plus gros des sous-domaines. Cela représente près d'un quart du temps d'exécution total.

Remarque 5.4

Une spécificité de *code_aster* fait que la résolution sur les semelles peut prendre du temps. Les valeurs des conditions de Dirichlet au niveau des interfaces des semelles changent à chaque appel de la fonction. Or il n'est pas possible, dans *code_aster*, de factoriser au préalable l'opérateur de rigidité associé sans fournir les valeurs des conditions imposées de Dirichlet. Une fois l'opérateur de rigidité factorisé, les valeurs conditions de Dirichlet ne peuvent pas être modifiées. Ainsi la résolution sur les semelles sous un chargement de Dirichlet à l'interface est effectuée avec une

re-factorisation à chaque appel de la fonction.

Quelques pistes d'amélioration sont à noter pour gagner en temps de calcul :

- éviter les déséquilibres de tailles des sous-domaines.
- pouvoir factoriser dans *code_aster* les opérateurs de rigidité des semelles pour des problèmes à chargement de Dirichlet
- mettre en place une gestion dynamique de la mémoire dans *code_aster* : *Aster HPC* dans le projet *PAMSIM*.

4 Quelques applications industrielles

4.1 Liaison de lanceur spatial

L'objectif de cet exemple est d'illustrer le comportement d'une structure sous deux cycles de charge / décharge. Nous nous intéressons au comportement d'une structure 3D industrielle représentant une liaison simplifiée de lanceur spatial mettant en jeu des interfaces de type contact et pré-charge. La structure est décomposée en 30 sous-domaines reliés par 53 interfaces. La description des sous-domaines et des interfaces est faite à l'annexe B section 1.1. Le modèle global possède près de 3 millions de degrés de liberté. La structure est encastree à une extrémité pendant qu'un effort de traction est imposé à l'autre. Une condition de symétrie bloque le déplacement normal à la face décrite à la figure B.1. 17 pas de chargements sont pris en compte dans cette simulation. Le premier consiste à pré-charge les 3 boulons au travers des 3 interfaces de pré-charges. Ensuite deux cycles de charge / décharge sont appliqués à la structure (Figure 5.12). Les paramètres globaux du modèle sont décrits au tableau 5.4.

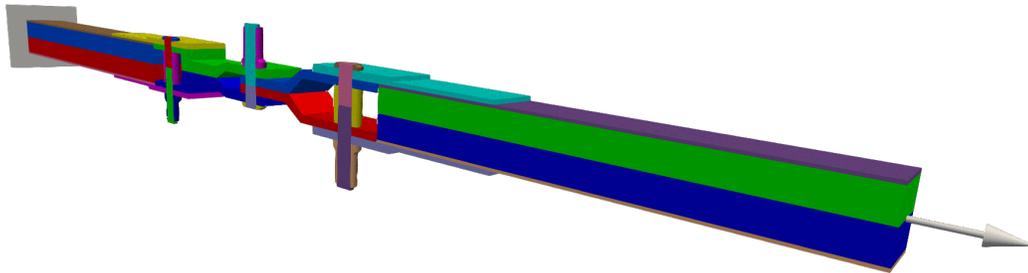


FIGURE 5.11 – Description de la bride sous-structurée

La figure 5.13 montre l'évolution de l'indicateur d'erreur Latin au cours des itérations. Bien que l'approche multi-échelle apporte une franche accélération aux toutes premières itérations, son effet s'estompe rapidement. Les nombreuses non-linéarités de contact en plus de la prise en compte de nombreux pas de chargement ralentissent la convergence : plus de 120 itérations sont nécessaires afin d'atteindre un critère d'erreur de l'ordre de 10^{-5} .

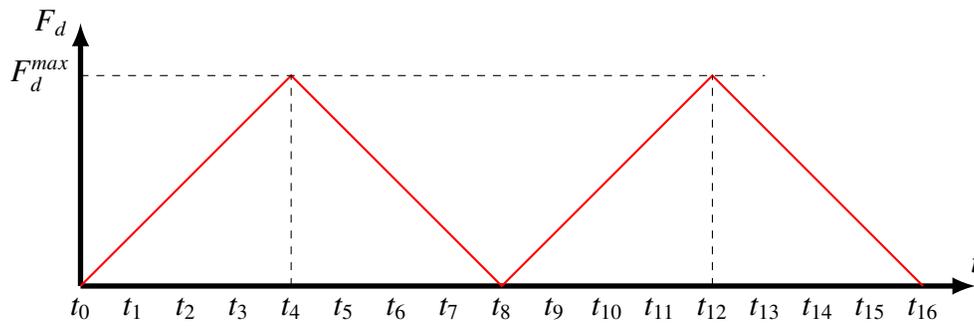


FIGURE 5.12 – Histoire du chargement

Paramètres	Valeur
Module de Young E	100 GPa
Coefficient de Poisson ν	0.3
F_d^{max}	10 MPa
pré-charge (gauche - milieu - droite)	0.3 - 0.2 - 0.4 mm
Δt	1s
ddls	3 millions

TABLE 5.4 – Paramètres

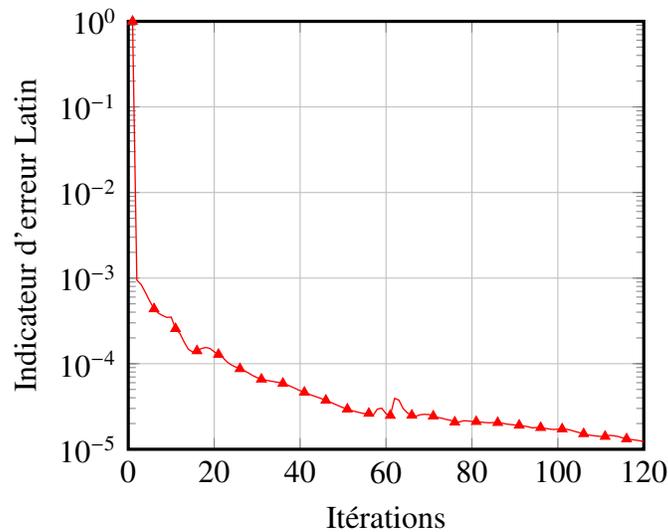


FIGURE 5.13 – Évolution de l'indicateur Latin

Sur la figure 5.14, la pré-charge des boulons peut être observée. Les tiges des vis sont bien mises en tension alors que les plaques environnantes sont soumises à des contraintes de compression.

La contrainte de Von Mises aux pas de chargement t_4 et t_{12} est représentée à la figure 5.15. Du glissement est observé entre les plaques et cela vient provoquer du cisaillement dans les tiges des

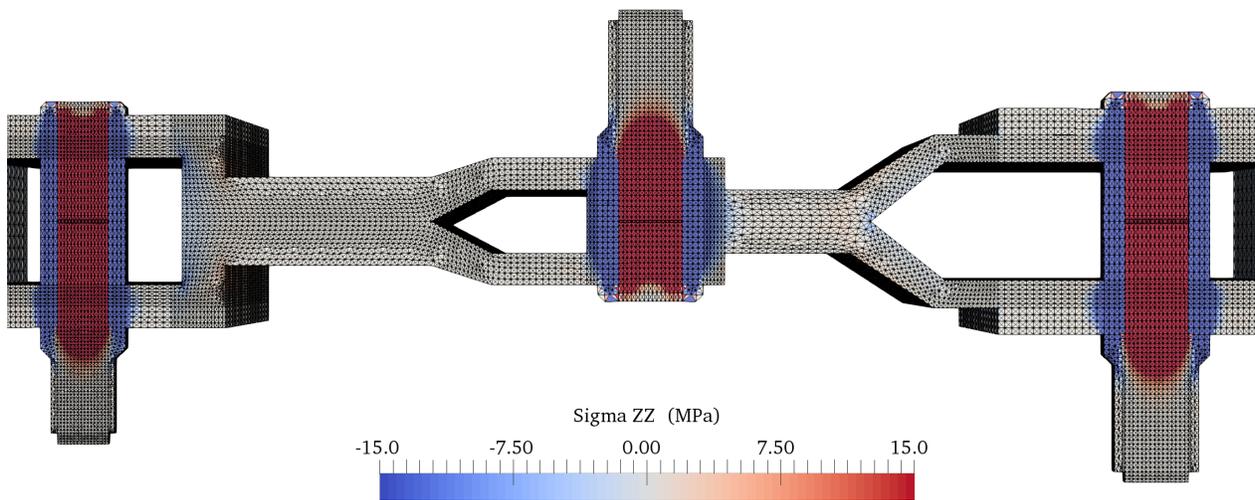


FIGURE 5.14 – Pré-charge des boulons

vis. La structure est plus sollicitée au 12^e pas de chargement qu'au 4^e à cause d'un glissement plus élevé entre les plaques qui vient cisailer plus fortement les tiges des vis.

En représentant l'évolution du déplacement longitudinal moyen en bout de structure en fonction du chargement, nous observons très nettement le glissement cumulé dans la structure. Au cours des charges, du glissement apparaît et celui-ci persiste lors des décharges. Nous n'avons simulé que deux cycles de chargement, cependant si d'autres cycles avaient été pris en compte, nous aurions observé une stabilisation de la solution vers un glissement cumulé limite.

4.2 Cas test industriel

4.2.1 Sous-structuration initiale

Nous étudions maintenant le comportement d'une structure boulonnée aéronautique dont la fonction est de lier les ailes au fuselage de l'avion. Cette structure a été fournie par AGI et nous permet de montrer la faisabilité d'une re-décomposition des sous-domaines pour atteindre un meilleur équilibrage en taille des sous-structures. Au total, cette structure contient 10 éléments boulonnés et est composée de 47 sous-domaines reliés entre eux au travers de 89 interfaces. Elle est représentée en coupe à la figure 5.17. La description complète des interfaces est détaillée en annexe B.2. La structure est encastree à ses extrémités droites et basses. Un effort dans la direction du sous-domaine S_2 est appliqué à l'extrémité gauche. Les éléments de visserie (tiges et écrous) sont en titane alors que le reste est composé d'aluminium.

Le chargement consiste en un cycle de charge / décharge à 9 pas de chargement (Figure 5.19)

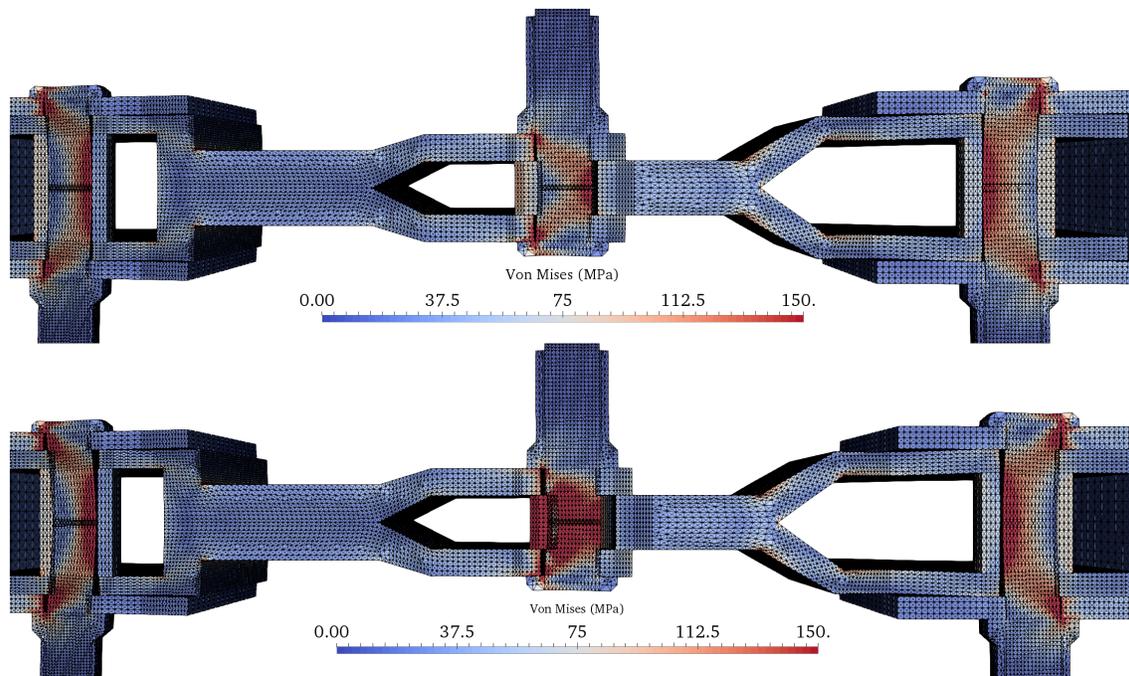


FIGURE 5.15 – Contrainte de Von Mises au pas de chargement t_4 et t_{12}

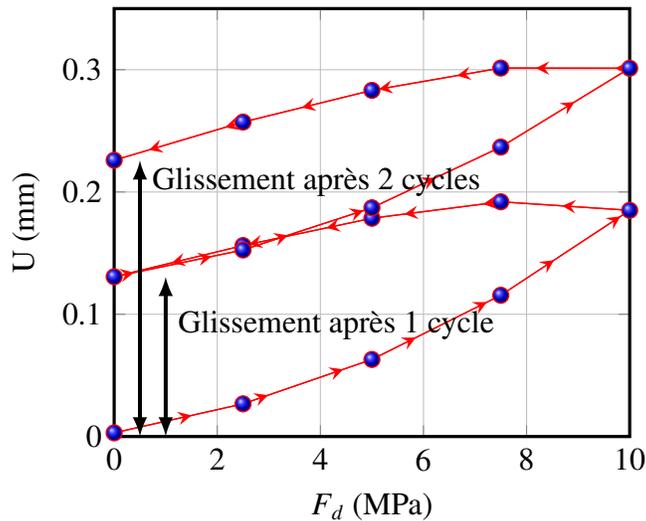


FIGURE 5.16 – Déplacement longitudinal moyen en bout de structure

dont le chargement maximal est de 5MPa, soit un effort résultant de 3.6kN. La figure 5.20 illustre l'état de contrainte de Von Mises au premier pas de chargement. Ici seule la pré-charge est appliquée et les éléments boulonnés sont bien chargés. La figure 5.21 présente l'évolution de la contrainte de Von Mises ainsi que la déformée au cours des différents pas de chargement. Nous constatons une liaison particulièrement plus sollicitée que les autres : l'élément boulonné le plus gros correspondant aux sous-domaines S_{43} , S_{44} , S_{45} et S_{46} .

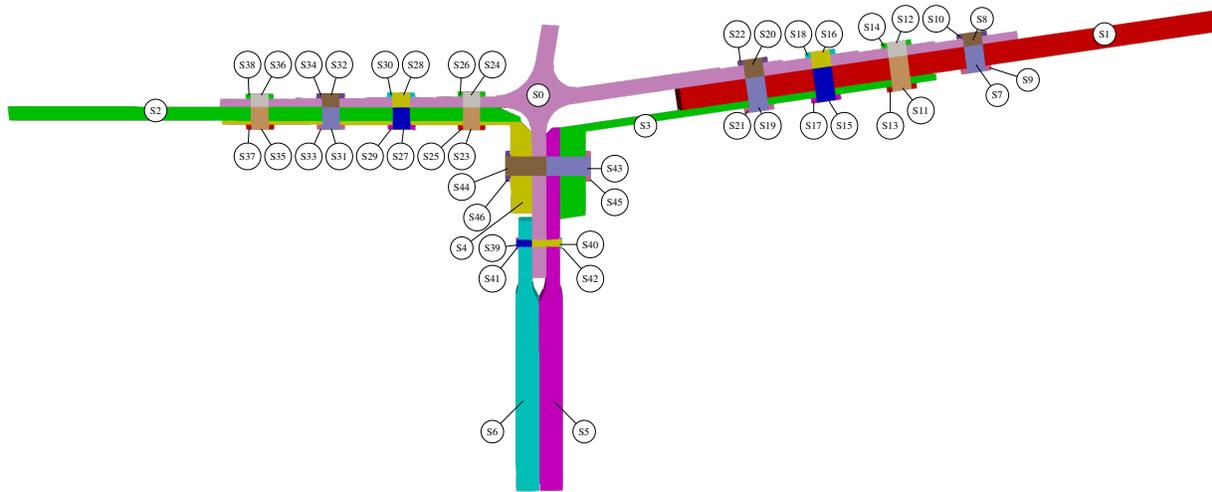


FIGURE 5.17 – Description des sous-domaines

Paramètres	Valeur
E_{alu}	70 GPa
ν_{alu}	0.3
E_{titane}	110 GPa
ν_{titane}	0.3
F_d^{max}	5 MPa
Pré-charge	0.005 mm
Jeu	0.005 mm
Δt	1s
ddls	300 000

FIGURE 5.18 – Paramètres

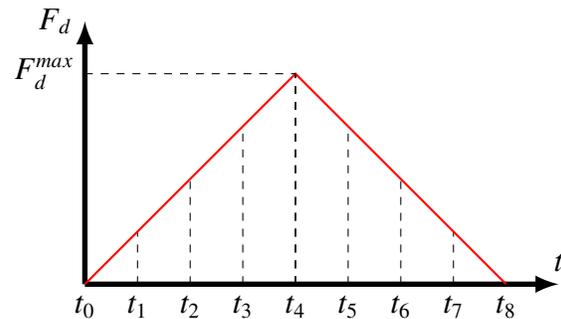
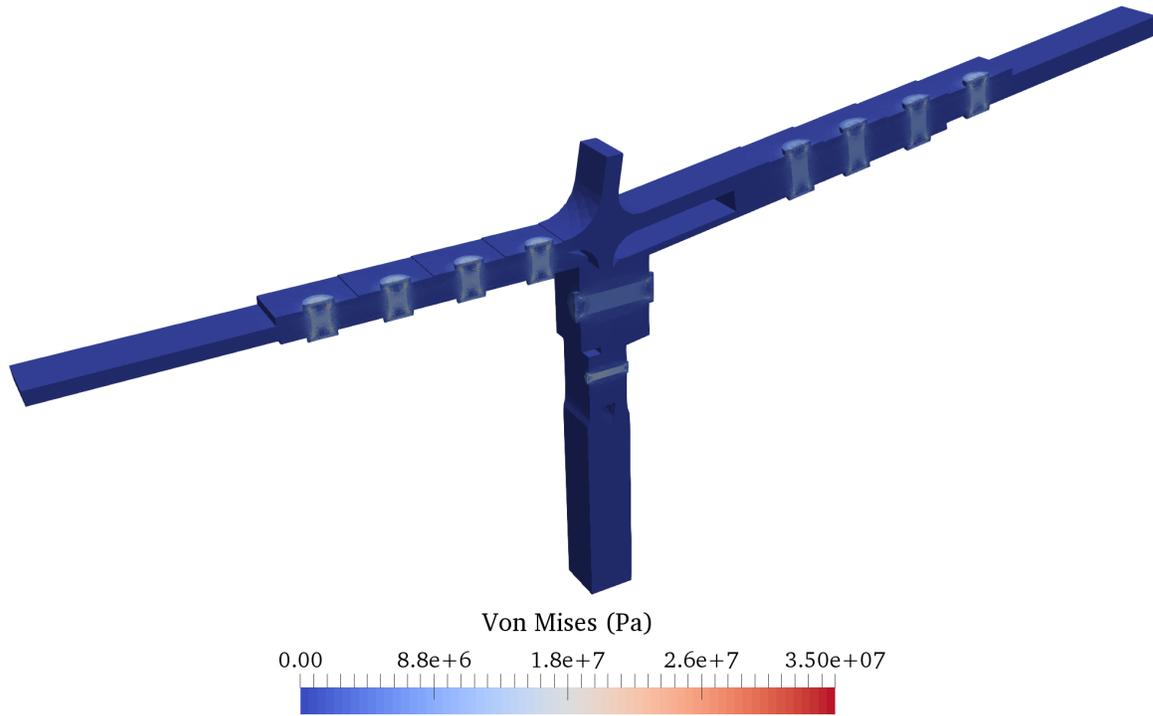


FIGURE 5.19 – Description du chargement

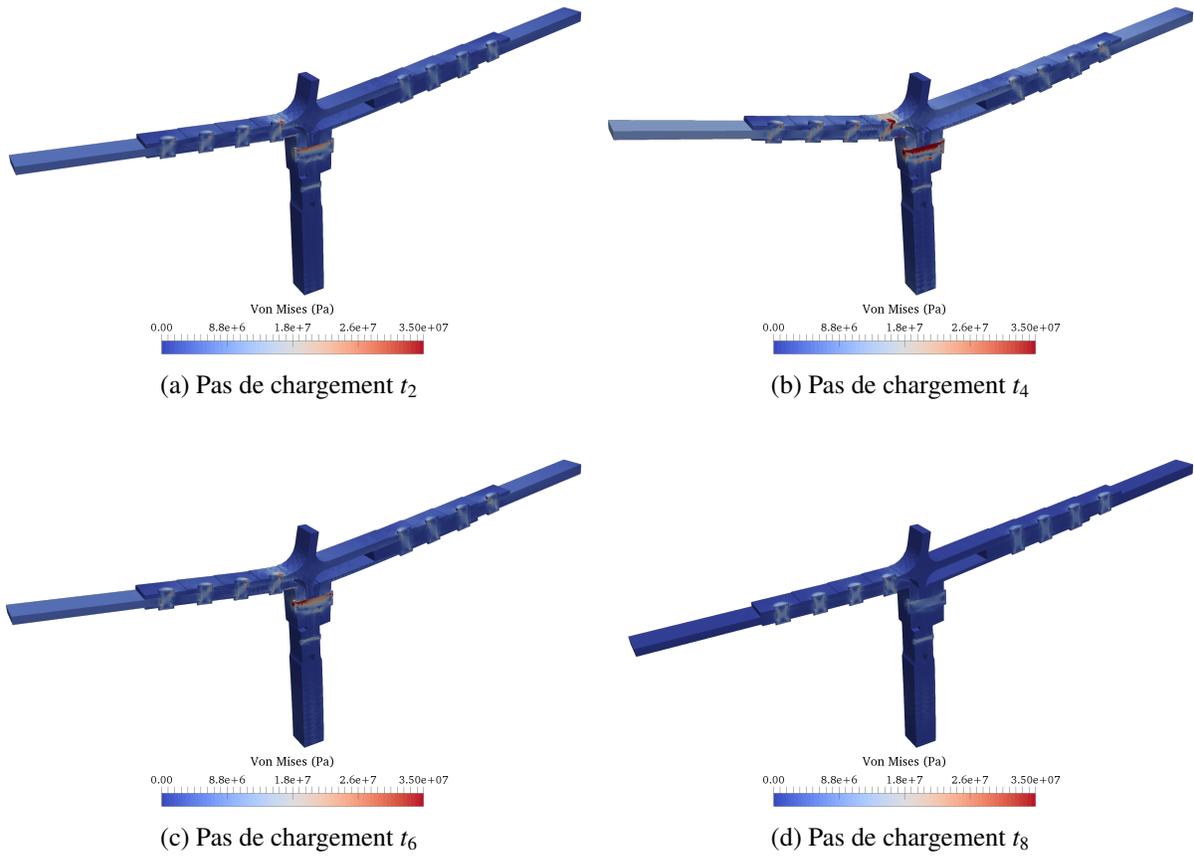
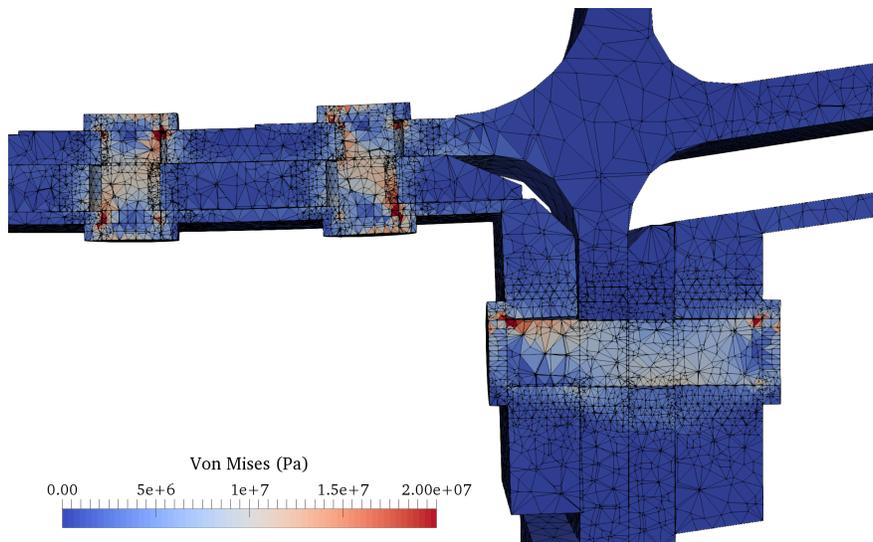
A la fin du cycle de chargement, du déplacement résiduel apparaît à cause du glissement cumulé dans les liaisons. La figure 5.22 se concentre sur la liaison la plus chargée. Du décollement est observé entre les sous-domaines S_0 et S_4 . Le glissement au niveau des éléments boulonnés est présent. L'interpénétration qui est visualisée entre les tiges et les plaques correspond au jeu initial non nul entre les pièces.

FIGURE 5.20 – Pré-charge au 1^e pas de chargement et animation

4.2.2 Re-sous-structuration pour équilibrer les sous-domaines

Pour la sous-structuration précédente qui correspond au maillage fourni par Airbus, le sous-domaine S_0 comporte 29 107 noeuds (Figure 5.17 : sous-domaine mauve). C'est le plus gros des sous-domaines et il est relié à 23 interfaces. Le temps de calcul total est donc directement piloté par ce sous-domaine. Nous proposons de venir re-décomposer les plus grosses sous-structures pour les re-équilibrer, à la fois en terme de degrés de liberté mais aussi en terme de nombre d'interfaces auxquelles elles sont reliées. Pour cela nous déterminons le nombre de noeuds initiaux de chaque sous-domaine. Les sous-domaines sont ensuite re-décomposés automatiquement à l'aide de *SCOTCH* pour que leur nombre final de noeuds soit de l'ordre de grandeur de celui du plus petit sous-domaine. La décomposition des sous-domaines est cependant majorée par un paramètre utilisateur M afin d'éviter d'en avoir un trop grand nombre. Le plus gros sous-domaine est alors décomposé en M nouvelles sous-structures. Les autres sous-domaines sont sous-structurés proportionnellement par rapport au plus gros. En notant n_S le nombre de noeuds du sous-domaine S et $a\%b$ le quotient de la division euclidienne de a par b , le calcul de la nouvelle décomposition des sous-domaine est précisée à l'algorithme 4.1.

Pour ce cas d'étude, en fixant la décomposition maximale M à 10 nouveaux sous-domaines, le nombre total augmente jusqu'à 69 (Figure 5.23). Nous avons ainsi effectué 70 itérations et le temps d'exécution total a été de 18 450 secondes contre 121 800 secondes pour le cas initial. Le fait d'avoir augmenté le nombre de sous-domaines, et donc utilisé plus de CPU, permet un gain en temps

FIGURE 5.21 – Contrainte de Von Mises et déformée ($\times 20$)FIGURE 5.22 – Carte de Von Mises et déformée $\times 100$

Algorithme 4.1 : Calcul de la nouvelle décomposition

```

for  $S \in \mathcal{E}$  do
   $\tilde{N}_S = n_S \% \min_{p \in \mathcal{E}}(n_p)$  ;           // Estimation de la nouvelle décomposition par sous-domaine
for  $S \in \mathcal{E}$  do
   $N_S = \tilde{N}_S \times M \% \max_{p \in \mathcal{E}}(\tilde{N}_p)$  ;       // Adaptation en fonction de la décomposition maximale
  
```

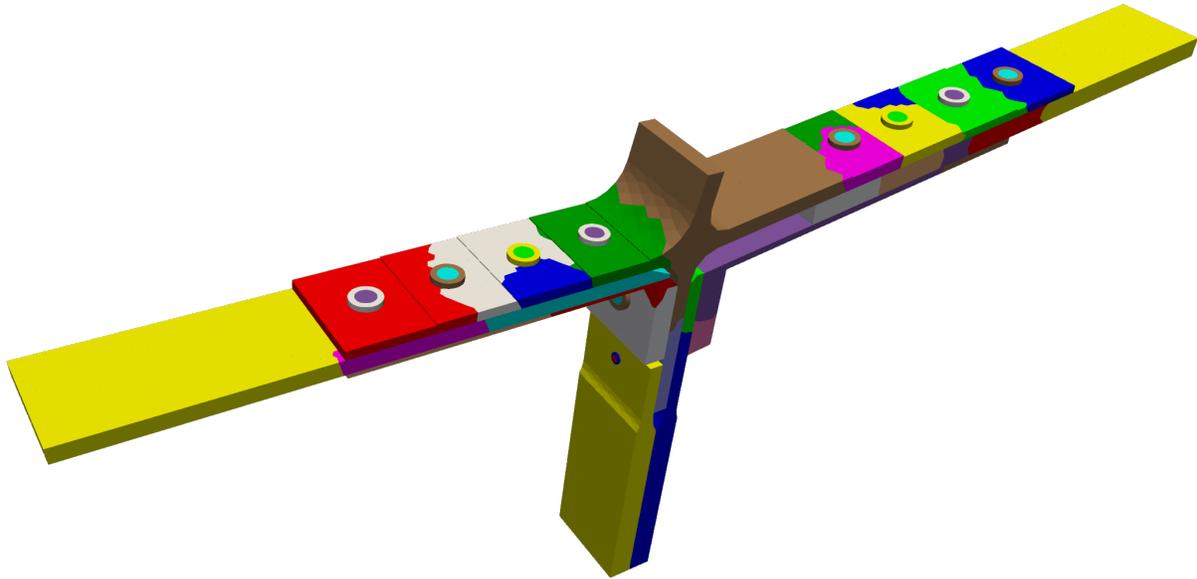


FIGURE 5.23 – Re-décomposition de la structure

de calcul de près **6,6**. La figure 5.24 présente la comparaison de l'indicateur d'erreur Latin. Le multi-échelle permet d'avoir une convergence initiale similaire entre les deux sous-structurations (jusqu'à 15 itérations). Cependant la convergence mono-échelle pour les itérations suivantes est moins performante pour le cas où la structure a été re-décomposée. Cette difficulté s'explique par le fait que *SCOTCH* a sous-structuré le sous-domaine initial S_0 en sous-domaines disjoints. La figure 5.25 représente la sous-structuration du sous-domaine S_0 en 10 nouvelles sous-structures. On note que deux nouveaux sous-domaines sont éclatés en parties disjointes : les sous-domaines en bleu et cyan. La boîte englobante ne permet pas de représenter correctement les dimensions réelles des sous-domaines et la pondération automatique des directions de recherche n'est plus adaptée. Cependant, cette méthode est robuste, puisque même avec une telle sous-structuration, elle converge.

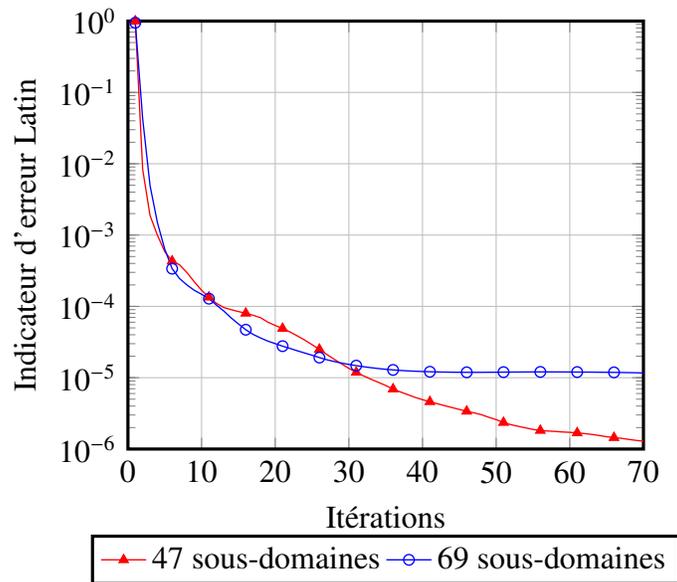
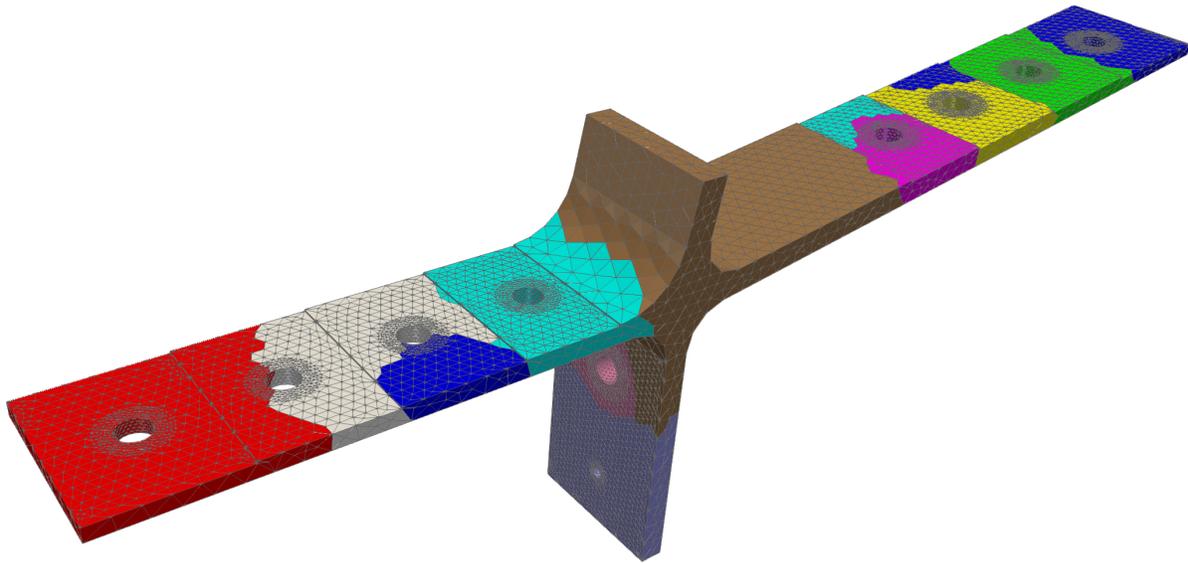


FIGURE 5.24 – Comparaison de l'indicateur d'erreur Latin

FIGURE 5.25 – Décomposition du sous-domaine S_0

5 Bilan

Au sein de ce chapitre nous avons abordé des questions relatives à l'implémentation de la méthode non-intrusive, en particulier à travers l'utilisation de *code_aster*. Des spécificités concernant le nommage ou la gestion mémoire des concepts ont nécessité des développements numériques

particuliers. L'outil développé permet de traiter facilement une grande variété de géométries. La construction automatique des interfaces et des semelles permet de minimiser le nombre de manipulations au sein de la phase de mise en données.

Sur des cas tests académiques nous avons mis en évidence un très bon *speed-up* (section 3.1). Nous avons présenté l'accélération à la fois pour une approche mono-échelle et une approche multi-échelle extensible.

La section 3.2 illustre le comportement de la méthode dans un cas fortement non-linéaire avec 100 sous-domaines et plus de 400 interfaces de contact. Une première configuration pour un coefficient de frottement faible de 0.1 et une configuration plus extrême avec un coefficient de frottement de 0.6 ont été simulées.

Deux simulations sur des structures industrielles ont été présentées à la section 4. La première concerne une liaison boulonnée de lanceur spatial. Trois boulons sont pris en compte sous un chargement cyclique qui illustre à la fin le glissement résiduel des différents éléments en contact. La seconde simulation représente une liaison boulonnée d'Airbus liant les ailes au fuselage. Cette fois-ci 10 boulons sont considérés. La sous-structuration étant déséquilibrée, un sous-domaine est beaucoup plus gros que les autres et devient limitant en terme de temps de résolution. Un outil de re-sous-structuration automatique a été créé pour rééquilibrer automatiquement les sous-domaines. Nous avons ainsi pu diminuer significativement le temps de calcul. La sous-structuration automatique amène encore quelques difficultés de convergence et un pilotage plus précis du décomposeur pourrait éviter ces quelques difficultés.

Conclusion

Le développement de méthodes de calcul adaptées aux ressources informatiques actuelles permet de gérer des problèmes de plus en plus complexes et de taille de plus en plus grande. Cependant, ces méthodes restent encore trop restreintes à la communauté scientifique académique. Le transfert vers le monde industriel n'est pas encore effectué. En particulier les méthodes de décomposition de domaine ne sont actuellement pas développées au sein des différents codes de calculs industriels. Nous nous sommes intéressés à la méthode Latin dont l'application à la sous-structuration conduit à une méthode de décomposition de domaine mixte. Les directions de recherche de la méthode introduisent des conditions de Robin sur les interfaces des sous-domaines. La prise en compte de ces conditions limites particulières n'est pas développée dans les codes industriels de mécanique. L'objectif de ces travaux de thèse a été de proposer une implémentation astucieuse des conditions de Robin afin d'éviter tout développement interne dans le code industriel *Code_Aster*.

Le [premier chapitre](#) a dressé un état de l'art sur les méthodes de décomposition de domaine et leurs adaptations pour gérer des interfaces de contact. Bien que de nombreux travaux se soient intéressés au cadre élastique linéaire, le traitement de problèmes non-linéaires de contact n'est que peu abordé. Les travaux les plus aboutis pour du contact frottant sont proposés par [[Dostál et al., 2010](#), [Dostál and Kučera, 2010](#)] pour les approches de type FETI.

Le [deuxième chapitre](#) a présenté l'implémentation non-intrusive de la méthode Latin mono-échelle. Les conditions de Robin sur les bords des sous-domaines, introduites par les directions de recherche mixtes, sont gérées par une impédance d'interface particulière. La direction de recherche de l'étape linéaire de la méthode Latin est désormais choisie comme étant non-locale par l'ajout d'une couche d'éléments aux interfaces des sous-domaines. Cela permet en particulier de traiter les conditions de Robin de l'étape linéaire par un assemblage de rigidité implémenté dans tout code industriel. Ce chapitre s'est intéressé seulement au cas d'interface parfaite. Des exemples 2D et 3D ont illustré la faisabilité de l'implémentation non-intrusive.

Ces travaux ont aussi été l'occasion de ré-écrire la méthode Latin en décomposition de domaine. L'étape linéaire a été étudiée dans le [troisième chapitre](#). Une formulation discrète a été proposée permettant une ré-écriture plus simple de l'approche multi-échelle. En particulier nous avons exhibé les expressions des opérateurs macroscopiques et déduit les propriétés algébriques des différents espaces microscopiques et macroscopiques (section 2). Nous avons expliqué qu'une approche multi-échelle visant à imposer une continuité globale des déplacements d'interface ne

permet pas de vérifier l'extensibilité de la méthode (section 4). Nous nous sommes aussi intéressés à inclure les itérations Latin au sein d'un solveur de Krylov (section 5). Les solutions sont donc cherchées dans des espaces de Krylov dont les propriétés assurent une convergence rapide. Ainsi cette accélération remplace la relaxation classique portant seulement sur deux itérations successives. Désormais, la solution d'une itération est optimisée par rapport à l'ensemble de celles déjà calculées. Pour des cas d'interfaces parfaites cette accélération de convergence apporte un gain particulièrement important.

L'étape locale de la méthode Latin a été détaillée au sein du [quatrième chapitre](#). Une description précise de la résolution du problème non-linéaire de contact a été proposée (section 2) : de l'obtention des indicateurs nodaux de contact et frottement à la résolution explicite du problème. Nous avons défini une interface de contact particulière pour imposer des pré-charges aux assemblages (section 3). La résolution des problèmes aux interfaces de contact nécessite de simplifier l'opérateur non-intrusif de direction de recherche. Le découplage des noeuds d'interface ainsi que le découplage des directions normales et tangentielles permet d'obtenir une solution explicite au problème de contact. Différents cas-test intermédiaires ont permis de valider les implémentations des interfaces de contact avant d'appliquer la méthode à une structure pré-chargée industrielle (section 4).

Au [cinquième chapitre](#) nous avons abordé l'implémentation informatique en détaillant quelques spécificités liées à *Code_Aster* (section 1 et section 2). Des résultats illustrant les performances sur des cas-tests académiques sont présentés à la section 3. Une étude du *speed-up* est réalisée pour un cas de statique à près de 25 millions de degrés de liberté. Une structure polycristalline mettant en jeu de très nombreuses interfaces de contact est aussi simulée et nous permet d'illustrer des performances en temps de calcul. La section 4 est consacrée à des applications industrielles. Une première structure représente une bride d'assemblage pour laquelle deux cycles de chargement sont simulés. La seconde, correspondant à un assemblage liant le fuselage aux ailes d'un avion, inclus 10 éléments boulonnés et de nombreuses interfaces de contact frottant. En particulier nous avons montré la possibilité de re-structuration automatique des pièces les plus grosses pour re-équilibrer la taille des sous-domaines entre eux.

Plus globalement ces travaux de thèse ont démontré la faisabilité de la mise en oeuvre non-intrusive. La gestion des conditions de Robin par une impédance d'interface non-locale peut être implémentée à moindre coût dans les codes éléments finis industriels. La philosophie de la Latin est particulièrement bien adaptée aux problèmes de contact frottant. Même avec de nombreuses non-linéarités de contact, la robustesse de la méthode Latin permet de traiter assez facilement une grande variété de structures.

Plusieurs perspectives à ces travaux peuvent être étudiées.

A court terme Le choix d'une direction de recherche de descente écrite en déplacement pour une étude quasi-statique permettrait de découpler temporellement les problèmes de l'étape linéaire. Ainsi une résolution multi seconds-membres sur les sous-domaines contenant les différents pas de chargement diminuerait efficacement les temps de calcul. La dépendance temporelle serait alors prise en compte au niveau de l'étape locale. La résolution multi seconds-membres n'étant pas implémentée dans *code_aster*, il n'a pu être possible d'en tester les performances.

EDF R&D développe une version HPC de *code_aster* : *aster HPC* au sein du projet PAM-SIM (PARallélisme Massif en Simulation numérique pour la MEcanique). L'objectif est de supprimer le superviseur pour intégrer une surcouche *cython/c++* afin de faire le lien entre l'environnement python et l'environnement fortran. Cette version de *Code_Aster* sera plus efficace et la gestion de la mémoire totalement dynamique. Un travail en collaboration avec l'équipe de développement permettrait de transférer l'outil développé pendant cette thèse sous *aster HPC* et ainsi de gagner en performance.

A moyen terme L'accélération de type Krylov pour les études avec des interfaces parfaites a montré de très bon résultats et des gains conséquents en terme d'itérations à convergence. Il s'agirait de poursuivre cette piste et de l'adapter pour être capable de gérer des problèmes à interface de contact.

Une gestion plus robuste des sous-structurations automatiques doit aussi être recherchée. En s'inspirant des travaux sur les méthodes de Schwarz [Spillane et al., 2011a, Spillane et al., 2011b] et de FETI [Spillane and Rixen, 2013] pour la définition automatique d'espaces grossiers, il doit être possible de gagner en robustesse dans la méthode Latin. En plus de chercher des modes macroscopiques adaptés, cela permettrait pour des cas industriels, de re-sous-structurer automatiquement les sous-domaines sans se préoccuper de la forme des interfaces.

A plus long terme Les études HPC requièrent plusieurs centaines, voire plusieurs milliers de coeurs. Sur ces nombreux CPU, la probabilité d'une défaillance informatique sur l'un d'eux n'est pas nulle et des techniques de gestion des défaillances doivent être mises en place. En particulier, des approches asynchrones [Magoulès et al., 2015, Magoulès and Gbikpi-Benissan, 2016] peuvent aider. La philosophie consiste à ne pas attendre, à la fin de chaque itération, la transmission des données des voisins mais de poursuivre à partir des données non mises à jour. Ainsi en cas de défaillance d'un processus, les autres sous-domaines poursuivent leurs itérations à partir des données du processus défaillant juste avant la panne. Cela permet aussi dans le cas de sous-domaines non équilibrés en taille de ne pas avoir à synchroniser les données et ainsi de ne pas faire attendre les sous-domaines les plus petits.

Précisions sur le traitement du contact

1 Démonstration

1.1 Cas du contact normal

On souhaite démontrer ici les équivalences suivantes :

$$\begin{aligned} \text{Contact} &\Leftrightarrow C_n \leq 0 \\ \text{Décollement} &\Leftrightarrow C_n > 0 \end{aligned} \tag{A.1}$$

Nous effectuons l'hypothèse que les directions de recherche normales k_n^+ sont telles que :

$$\forall \Gamma_{ij} \in \mathcal{G}, k_n^{+ij} = k_n^{+ji} \tag{A.2}$$

Cela implique en particulier que l'on peut écrire :

$$B_n k_n^+ = k_n^+ B_n \tag{A.3}$$

1.1.1 Décollement

Décollement $\Rightarrow C_n > 0$ On fait l'hypothèse que nous sommes dans un cas de décollement. Par conséquent la relation de non interpénétration avec jeu initial est une inégalité stricte.

$$\begin{aligned} B_n \widehat{W}_n^{t+\Delta t} + j_n > 0 &\Rightarrow C_n + \Delta t B_n k_n^{+-1} \underbrace{\widehat{F}_n^t}_{=0 \text{ car décollement}} > 0 \\ &\Rightarrow C_n > 0 \end{aligned} \tag{A.4}$$

Ainsi on a bien :

$$\boxed{\text{Décollement} \Rightarrow C_n > 0} \tag{A.5}$$

$C_n > 0 \Rightarrow$ **Décollement** On suppose que $C_n > 0$.

On a montré que la relation de non-interpénétration s'écrit :

$$B_n \widehat{W}_n^{t+\Delta t} + j_n = C_n + \Delta t \mathbf{B}_n k_n^{+-1} \widehat{F}_n^{t+\Delta t} = C_n + \Delta t k_n^{+-1} \mathbf{B}_n \widehat{F}_n^{t+\Delta t} \quad (\text{A.6})$$

Or on a supposé que $C_n > 0$ et quelque soit le cas on vérifie $B_n \widehat{F}_n^{t+\Delta t} \geq 0$. Ainsi :

$$\begin{cases} C_n > 0 \\ B_n \widehat{F}_n^{t+\Delta t} \geq 0 \end{cases} \Rightarrow C_n + \Delta t k_n^{+-1} \mathbf{B}_n \widehat{F}_n^{t+\Delta t} > 0 \Rightarrow B_n \widehat{W}_n^{t+\Delta t} + j_n > 0 \text{ (Décollement)} \quad (\text{A.7})$$

D'où l'implication :

$$\boxed{C_n > 0 \Rightarrow \text{Décollement}} \quad (\text{A.8})$$

Ainsi on vérifie bien :

$$\boxed{\text{Décollement} \Leftrightarrow C_n > 0} \quad (\text{A.9})$$

1.1.2 Contact

Contact $\Rightarrow C_n \leq 0$ On suppose que l'on est dans un cas de contact. Par conséquent la relation de non-interpénétration est une égalité :

$$\begin{cases} C_n + \Delta t k_n^{+-1} B_n \widehat{F}_n^{t+\Delta t} = 0 \\ B_n \widehat{F}_n^{t+\Delta t} \geq 0 \end{cases} \Rightarrow C_n \leq 0 \quad (\text{A.10})$$

Ainsi :

$$\boxed{\text{Contact} \Rightarrow C_n \leq 0} \quad (\text{A.11})$$

$C_n \leq 0 \Rightarrow$ **Contact** On suppose tout d'abord que $C_n < 0$. D'où :

$$C_n + \Delta t k_n^{+-1} B_n \widehat{F}_n^{t+\Delta t} \geq 0 \Rightarrow B_n \widehat{F}_n^{t+\Delta t} > 0 \quad (\text{A.12})$$

La relation de Signorini impose par conséquent que $B_n \widehat{W}_n^{t+\Delta t} + j_n = 0$ et donc contact au noeud.

Dans le cas où $C_n = 0$, on peut écrire que $\Delta t k_n^{+-1} B_n \widehat{F}_n^{t+\Delta t} = B_n \widehat{W}_n^{t+\Delta t} + j_n \geq 0$. Ainsi la condition d'exclusion devient :

$$(B_n \widehat{F}_n^{t+\Delta t}) (\Delta t k_n^{+-1} B_n \widehat{F}_n^{t+\Delta t}) = 0 \quad (\text{A.13})$$

Ce qui conduit inévitablement à ce que $B_n \widehat{F}_n^{t+\Delta t} = 0$ soit encore $B_n \widehat{W}_n^{t+\Delta t} + j_n = 0$. Ainsi le contact est obtenu et dans le cas où $C_n = 0$ le contact a lieu sans effort normal.

On a bien montré :

$$\boxed{C_n \leq 0 \Rightarrow \text{Contact}} \quad (\text{A.14})$$

Ainsi on vérifie bien :

$$\boxed{\text{Contact} \Leftrightarrow C_n \leq 0} \quad (\text{A.15})$$

1.2 Cas du frottement

On souhaite démontrer ici les équivalences suivantes :

$$\begin{aligned} \text{Adhérence} &\Leftrightarrow \|G_\tau\| < \mu|\widehat{F}_n^{t+\Delta t}| \\ \text{Glissement} &\Leftrightarrow \|G_\tau\| \geq \mu|\widehat{F}_n^{t+\Delta t}| \end{aligned} \quad (\text{A.16})$$

Adhérence $\Rightarrow \|G_\tau\| < \mu|\widehat{F}_n^{t+\Delta t}|$ On suppose être dans un cas d'adhérence, ainsi on vérifie $\|\widehat{F}_\tau^{t+\Delta t}\| < \mu|\widehat{F}_n^{t+\Delta t}|$ et $B_\tau \widehat{W}_\tau^{t+\Delta t} = 0$. On rappelle que $\widehat{F}_\tau^{t+\Delta t} = G_\tau + B_\tau^T \beta$.

Supposons que $\beta \neq 0$. Ainsi $B_\tau \widehat{W}_\tau^{t+\Delta t} = B_\tau k_\tau^{+1} B_\tau^T \beta \neq 0$ ce qui est absurde car on est dans un cas d'adhérence. D'où β est nul et donc $\widehat{F}_\tau^{t+\Delta t} = G_\tau$.

Par conséquent $\|\widehat{F}_\tau^{t+\Delta t}\| < \mu|\widehat{F}_n^{t+\Delta t}| \Rightarrow \|G_\tau\| < \mu|\widehat{F}_n^{t+\Delta t}|$.

Ainsi :

$$\boxed{\text{Adhérence} \Rightarrow \|G_\tau\| < \mu|\widehat{F}_n^{t+\Delta t}|} \quad (\text{A.17})$$

Remarque A.0

Par contraposé on montre que :

$$\boxed{\|G_\tau\| \geq \mu|\widehat{F}_n^{t+\Delta t}| \Rightarrow \text{Glissement}} \quad (\text{A.18})$$

Glissement $\Rightarrow \|G_\tau\| \geq \mu|\widehat{F}_n^{t+\Delta t}|$ On suppose être dans un cas de glissement. On vérifie ainsi $\|\widehat{F}_\tau^{t+\Delta t}\| = \mu|\widehat{F}_n^{t+\Delta t}|$ et $\exists \lambda > 0$, $B_\tau \widehat{W}_\tau^{t+\Delta t} = -\lambda (B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_\tau^{t+\Delta t}$.

On obtient alors une relation entre β et $\widehat{F}_\tau^{t+\Delta t}$:

$$(B_\tau B_\tau^T)^{-1} B_\tau \widehat{F}_\tau^{t+\Delta t} = - (B_\tau k_\tau^{+1} B_\tau^T) \frac{\beta}{\lambda} \quad (\text{A.19})$$

On en déduit que $\widehat{F}_\tau^{t+\Delta t}$ et $B_\tau^T \beta$ sont colinéaires et de sens contraire. Ainsi :

$$\exists \rho > 0, B_\tau^T \beta = -\rho \widehat{F}_\tau^{t+\Delta t} \quad (\text{A.20})$$

D'où :

$$G_\tau = (1 + \rho) \widehat{F}_\tau^{t+\Delta t} \Rightarrow \|G_\tau\| \geq \mu|\widehat{F}_n^{t+\Delta t}| \quad (\text{A.21})$$

Ainsi :

$$\boxed{\text{Glissement} \Rightarrow \|G_\tau\| \geq \mu|\widehat{F}_n^{t+\Delta t}|} \quad (\text{A.22})$$

 **Remarque A.1**

Par contraposé on montre aussi que :

$$\|G_\tau\| < \mu|\widehat{F}_n^{t+\Delta t}| \Rightarrow \text{Adhérence} \quad (\text{A.23})$$

Finalement les équivalences sont bien vérifiées :

$$\begin{array}{l} \text{Adhérence} \quad \Leftrightarrow \|G_\tau\| < \mu|\widehat{F}_n^{t+\Delta t}| \\ \text{Glissement} \quad \Leftrightarrow \|G_\tau\| \geq \mu|\widehat{F}_n^{t+\Delta t}| \end{array} \quad (\text{A.24})$$

Description des structures industrielles

1 Description des cas industriels

1.1 Bride

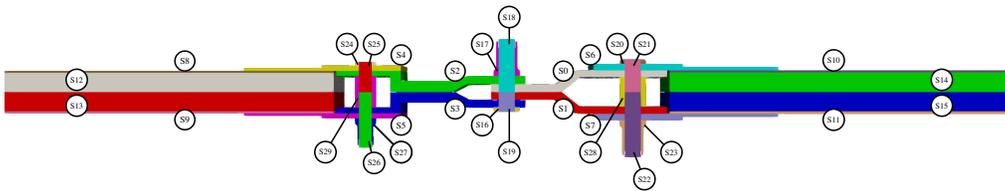


FIGURE B.1 – Description des sous-domaines

Description des interfaces

Interface	SD	Type	Coefficient frottement	Jeu	Pre-charge
0	S0 / S1	Parfaite			
1	S0 / S2	Contact	0.3		
2	S0 / S6	Contact	0.3		
3	S0 / S18	Contact	0.3	0.05	
4	S0 / S21	Contact	0.3	0.05	
5	S0 / S28	Contact	0.3		
6	S1 / S3	Contact	0.3		
7	S1 / S7	Contact	0.3		
8	S1 / S19	Contact	0.3	0.05	
9	S1 / S22	Contact	0.3	0.05	

10	S1 / S28	Contact	0.3	
11	S2 / S3	Parfaite		
12	S2 / S4	Contact	0.3	
13	S2 / S17	Contact	0.3	
14	S2 / S18	Contact	0.3	0.05
15	S2 / S25	Contact	0.3	0.05
16	S2 / S29	Contact	0.3	
17	S3 / S5	Contact	0.3	
18	S3 / S16	Contact	0.3	
19	S3 / S19	Contact	0.3	0.05
20	S3 / S26	Contact	0.3	0.05
21	S3 / S29	Contact	0.3	
22	S4 / S8	Parfaite		
23	S4 / S24	Contact	0.3	
24	S4 / S25	Contact	0.3	0.05
25	S5 / S9	Parfaite		
26	S5 / S26	Contact	0.3	0.05
27	S5 / S27	Contact	0.3	
28	S6 / S10	Parfaite		
29	S6 / S20	Contact	0.3	
30	S6 / S21	Contact	0.3	0.05
31	S7 / S11	Parfaite		
32	S7 / S22	Contact	0.3	0.05
33	S7 / S23	Contact	0.3	
34	S8 / S12	Parfaite		
35	S9 / S13	Parfaite		
36	S10 / S14	Parfaite		
37	S11 / S15	Parfaite		
38	S12 / S13	Parfaite		
39	S14 / S15	Parfaite		
40	S16 / S19	Parfaite		
41	S17 / S18	Parfaite		
42	S18 / S19	Pré-charge		0.2
43	S20 / S21	Parfaite		
44	S21 / S22	Pré-charge		0.4
45	S21 / S28	Contact	0.3	0.05
46	S22 / S23	Parfaite		
47	S22 / S28	Contact	0.3	0.05
48	S24 / S25	Parfaite		
49	S25 / S26	Pré-charge		0.3
50	S25 / S29	Contact	0.3	0.05
51	S26 / S27	Parfaite		

52	S26 / S29	Contact	0.3	0.05
----	-----------	---------	-----	------

TABLE B.1 – Description des interfaces

1.2 Laison boulonnée Airbus

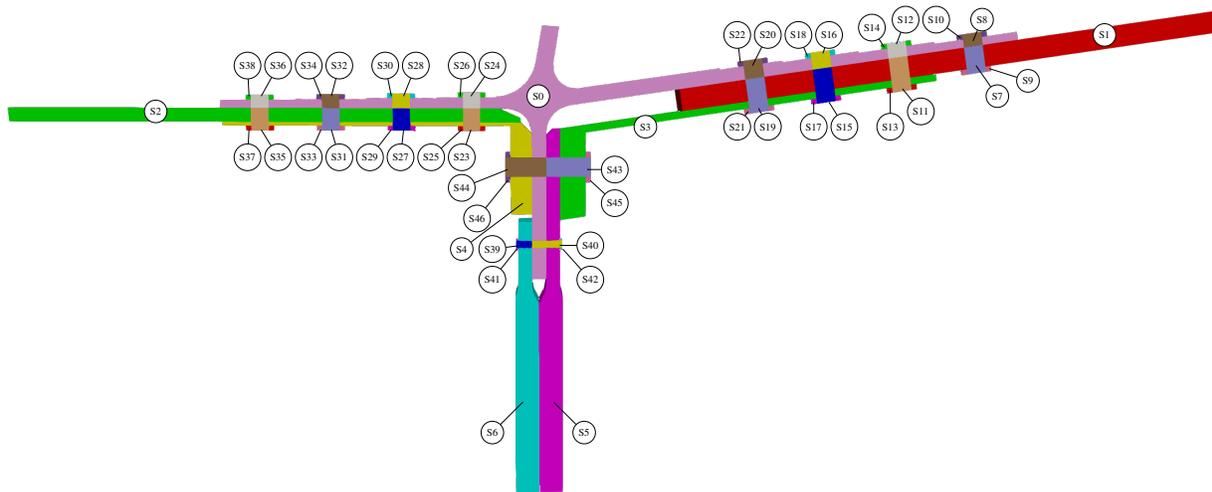


FIGURE B.2 – Description des sous-domaines

Interface	SD	Type	Coefficient frottement	Jeu	Pre-charge
0	S0 / S1	Contact	0.3		
1	S0 / S2	Contact	0.3		
2	S0 / S4	Contact	0.3		
3	S0 / S5	Contact	0.3		
4	S0 / S6	Contact	0.3		
5	S0 / S8	Contact	0.	0.0005	
6	S0 / S10	Contact	0.3		
7	S0 / S12	Contact		0.0005	
8	S0 / S14	Contact	0.3		
9	S0 / S16	Contact		0.0005	
10	S0 / S18	Contact	0.3		
11	S0 / S20	Contact		0.0005	

12	S0 / S22	Contact	0.3	
13	S0 / S24	Contact		0.0005
14	S0 / S26	Contact	0.3	
15	S0 / S28	Contact		0.0005
16	S0 / S30	Contact	0.3	
17	S0 / S32	Contact		0.0005
18	S0 / S34	Contact	0.3	
19	S0 / S36	Contact		0.0005
20	S0 / S38	Contact	0.3	
21	S0 / S40	Contact		0.0005
22	S0 / S44	Contact		0.0005
23	S1 / S3	Contact	0.3	
24	S1 / S7	Contact		0.0005
25	S1 / S9	Contact	0.3	
26	S1 / S11	Contact		0.0005
27	S1 / S15	Contact		0.0005
28	S1 / S19	Contact		0.0005
29	S2 / S4	Contact	0.3	
30	S2 / S23	Contact		0.0005
31	S2 / S27	Contact		0.0005
32	S2 / S31	Contact		0.0005
33	S2 / S35	Contact		0.0005
34	S3 / S5	Contact	0.3	
35	S3 / S11	Contact		0.0005
36	S3 / S13	Contact	0.3	
37	S3 / S15	Contact		0.0005
38	S3 / S17	Contact	0.3	
39	S3 / S19	Contact		0.0005
40	S3 / S21	Contact	0.3	
41	S3 / S43	Contact		0.0005
42	S3 / S45	Contact	0.3	
43	S4 / S23	Contact		0.0005
44	S4 / S25	Contact	0.3	
45	S4 / S27	Contact		0.0005
46	S4 / S29	Contact	0.3	
47	S4 / S31	Contact		0.0005
48	S4 / S33	Contact	0.3	
49	S4 / S35	Contact		0.0005
50	S4 / S37	Contact	0.3	
51	S4 / S44	Contact		0.0005
52	S4 / S46	Contact	0.3	
53	S5 / S6	Contact	0.3	
54	S5 / S40	Contact		0.0005

55	S5 / S42	Contact	0.3	
56	S5 / S43	Contact		0.0005
57	S6 / S39	Contact		0.0005
58	S6 / S41	Contact	0.3	
59	S7 / S8	Pré-charge		0.0005
60	S7 / S9	Parfaite		
61	S8 / S10	Parfaite		
62	S11 / S12	Pré-charge		0.0005
63	S11 / S13	Parfaite		
64	S12 / S14	Parfaite		
65	S15 / S16	Pré-charge		0.0005
66	S15 / S17	Parfaite		
67	S16 / S18	Parfaite		
68	S19 / S20	Pré-charge		0.0005
69	S19 / S21	Parfaite		
70	S20 / S22	Parfaite		
71	S23 / S24	Pré-charge		0.0005
72	S23 / S25	Parfaite		
73	S24 / S26	Parfaite		
74	S27 / S28	Pré-charge		0.0005
75	S27 / S29	Parfaite		
76	S28 / S30	Parfaite		
77	S31 / S32	Pré-charge		0.0005
78	S31 / S33	Parfaite		
79	S32 / S34	Parfaite		
80	S35 / S36	Pré-charge		0.0005
81	S35 / S37	Parfaite		
82	S36 / S38	Parfaite		
83	S39 / S40	Pré-charge		0.0005
84	S39 / S41	Parfaite		
85	S40 / S42	Parfaite		
86	S43 / S44	Pré-charge		0.0005
87	S43 / S45	Parfaite		
88	S44 / S46	Parfaite		

TABLE B.2 – Description des interfaces

Bibliographie

- [Abel and Shepard, 1979] Abel, J. F. and Shepard, M. S. (1979). An algorithm for multipoint constraints in finite element analysis. *International Journal for Numerical Methods in Engineering*, 14(3) :464–467.
- [Alart and Curnier, 1991] Alart, P. and Curnier, A. (1991). A mixed formulation for frictional contact problems prone to Newton like solution methods. *Computer Methods in Applied Mechanics and Engineering*, 92(3) :353–375.
- [Alart, 1997] Alart, P. P. (1997). Méthode de Newton généralisée en mécanique du contact. *Journal de Mathématiques Pures et Appliquées*, 76(1) :83–108.
- [Allix and Ladevèze, 1992] Allix, O. and Ladevèze, P. (1992). Interlaminar interface modelling for the prediction of delamination. *Composite Structures*, 22(4) :235–242.
- [Amestoy et al., 2001] Amestoy, P. R., Duff, I. S., Koster, J., and L’Excellent, J.-Y. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1) :15–41.
- [Annavarapu et al., 2015] Annavarapu, C., Settigast, R. R., Johnson, S. M., Fu, P., and Herbold, E. B. (2015). A weighted Nitsche stabilized method for small-sliding contact on frictional surfaces. *Comput. Methods Appl. Mech. Engrg.*, 283 :763–781.
- [Arrow K., 1958] Arrow K., Hurwicz L., U. H. (1958). *Studies in nonlinear programming*. Stanford University Press.
- [Arzt, 1994] Arzt, M. (1994). *Approche des phénomènes cycliques par la méthode à grand incrément de temps*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Avery and Farhat, 2009] Avery, P. and Farhat, C. (2009). The FETI family of domain decomposition methods for inequality-constrained quadratic programming : Application to contact problems with conforming and nonconforming. *Computer Methods in Applied Mechanics and . . .*, 198(21-26) :1673–1683.
- [Avery et al., 2004] Avery, P., Rebel, G., Lesoinne, M., and Farhat, C. (2004). A numerically scalable dual-primal substructuring method for the solution of contact problems—part I : the frictionless case. *Computer Methods in Applied Mechanics and Engineering*, 193(23) :2403–2426.

- [Badea and Krause, 2012] Badea, L. and Krause, R. (2012). One- and two-level Schwarz methods for variational inequalities of the second kind and their application to frictional contact. *Numerische Mathematik*, pages 573–599.
- [Badea and Lebon, 2015] Badea, L. and Lebon, F. (2015). Schwarz method for dual contact problems. *Computational and Applied Mathematics*.
- [Badea et al., 2003] Badea, L., Tai, X.-c., and Wang Junping (2003). Convergence rate analysis of a multiplicative Schwarz method for variational inequalities. *Siam J. Numer. Anal.*, 41(3) :1052–1073.
- [Badea and Wang, 1999] Badea, L. and Wang, J. (1999). An additive Schwarz method for variational inequalities. *Mathematics of Computation*, 69(232) :1341–1354.
- [Ballard, 1999] Ballard, P. (1999). A counter-example to uniqueness in quasi-static elastic contact problems with small friction. *International Journal of Engineering Science*, 37(2) :163–178.
- [Becker et al., 2003] Becker, R., Hansbo, P., and Stenberg, R. (2003). A finite element method for domain decomposition with non-matching grids. *ESAIM : Mathematical Modelling and Numerical Analysis*, 37(2) :209–225.
- [Bertsekas, 2014] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.
- [Bettinotti et al., 2014] Bettinotti, O., Allix, O., Perego, U., Oancea, V., and Malherbe, B. (2014). A fast weakly intrusive multiscale method in explicit dynamics. *International Journal for Numerical Methods in Engineering*, 100 :577–595.
- [Blanchard et al., 2017] Blanchard, M., Allix, O., Gosselet, P., and Desmeure, G. (2017). Mastering the convergence of the global-local non-invasive coupling technique in viscoplasticity. *Finite Element Analysis and Design*, x :submitted.
- [Blanze et al., 1996] Blanze, C., Champaney, L., Cognard, J.-Y., and Ladevèze, P. (1996). A modular approach to structure assembly computations : application to contact problems. *Engineering Computations*, 13(1) :15–32.
- [Blanzé et al., 2000] Blanzé, C., Champaney, L., and Vedrine, P. (2000). Contact problems in the design of a superconducting quadrupole prototype. *Engineering Computations*, 17(2) :136–153.
- [Boisse, 1987] Boisse, P. (1987). *Nouvel algorithme à grand incrément de temps pour le calcul des structures élastoplastiques*. PhD thesis, Université Pierre et Marie Curie.
- [Boucard, 1996] Boucard, P.-A. (1996). *Approche à grand incrément de temps en grandes transformations*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Boucard and Champaney, 2003] Boucard, P. A. and Champaney, L. (2003). A suitable computational strategy for the parametric analysis of problems with multiple contact. *International Journal for Numerical Methods in Engineering*, 57(9) :1259–1281.
- [Bourgat et al., 1989] Bourgat, J.-F., Glowinski, R., Le Tallec, P., and Vidrascu, M. (1989). Variational formulation and algorithm for trace operator in domain decomposition calculations. *Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, Domain Decomposition Methods, Philadelphia, PA*.

- [Burman and Fernández, 2014] Burman, E. and Fernández, M. A. (2014). An unfitted Nitsche method for incompressible fluid-structure interaction using overlapping meshes. *Computer Methods in Applied Mechanics and Engineering*, 279 :497–514.
- [Cai and Sarkis, 1999] Cai, X.-C. and Sarkis, M. (1999). A restricted additive Schwarz preconditioner for general sparse linear systems. *Siam journal on scientific computing*, 21(2) :792–797.
- [Cai and Widlund, 1993] Cai, X.-C. and Widlund, O. B. (1993). Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems. *SIAM Journal on Numerical Analysis*, 30(4) :936–952.
- [Champaney, 1996] Champaney, L. (1996). *Une nouvelle approche modulaire pour l'analyse d'assemblages de structures tridimensionnelles*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Chan and Mathew, 1994] Chan, T. F. and Mathew, T. P. (1994). Domain decomposition algorithms. *Acta numerica*, 3 :61–143.
- [Christensen, 2002] Christensen, P. W. (2002). A semi-smooth Newton method for elasto-plastic contact problems. *International journal of solids and structures*, 39(8) :2323–2341.
- [Clarke, 1990] Clarke, F. H. (1990). *Optimization and nonsmooth analysis*. SIAM.
- [Cocu et al., 1996] Cocu, M., Pratt, E., and Raous, M. (1996). Formulation and approximation of quasistatic frictional contact. *International Journal of Engineering Science*, 34(7) :783–798.
- [Cognard et al., 1999] Cognard, J., Ladeveze, P., and Talbot, P. (1999). A large time increment approach for thermo-mechanical problems. *Advances in Engineering Software*, 30(9-11) :583–593.
- [Cognard, 1989] Cognard, J.-Y. (1989). *Une Nouvelle approche des problèmes de plasticité et de viscoplasticité : la méthode à grand incrément de temps*. PhD thesis, Université Pierre et Marie Curie.
- [Cognard and Ladevèze, 1993] Cognard, J.-Y. and Ladevèze, P. (1993). A large time increment approach for cyclic viscoplasticity. *International Journal of Plasticity*, 9(2) :141–157.
- [Confalonieri et al., 2013a] Confalonieri, F., Cocchetti, G., Ghisi, A., and Corigliano, A. (2013a). A domain decomposition method for the simulation of fracture in polysilicon MEMS. *Microelectronics Reliability*, 53(8) :1045–1054.
- [Confalonieri et al., 2013b] Confalonieri, F., Corigliano, A., Dossi, M., and Gornati, M. (2013b). A domain decomposition technique applied to the solution of the coupled electro-mechanical problem. *International Journal for Numerical Methods in Engineering*, 93(2) :137–159.
- [Conn et al., 1991] Conn, A. R., Gould, N. I. M., and Toint, P. L. (1991). A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds. *SIAM Journal on Numerical Analysis*, 28(2) :545–572.
- [Curnier and Alart, 1988] Curnier, A. and Alart, P. (1988). A generalized Newton method for contact problems with friction. *Journal de mécanique théorique et appliquée*.
- [De Roeck and Le Tallec, 1991] De Roeck, Y.-H. and Le Tallec, P. (1991). Analysis and test of a local domain decomposition preconditioner. In *Fourth international symposium on domain decomposition methods for partial differential equations*, volume 4.

- [Desmeure, 2016] Desmeure, G. (2016). *Une stratégie de décomposition de domaine mixte et multiéchelle pour le calcul des assemblages*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Dohrmann, 2003] Dohrmann, C. R. (2003). A preconditioner for substructuring based on constrained energy minimization. *SIAM Journal on Scientific Computing*, 25(1) :246–258.
- [Dohrmann, 2007] Dohrmann, C. R. (2007). An approximate BDDC preconditioner. *Numerical Linear Algebra with Applications*, 14(2) :149–168.
- [Dostál, 1997] Dostál, Z. (1997). Box constrained quadratic programming with proportioning and projections. *SIAM J. Opt.*, 7(3) :871–887.
- [Dostál, 2006a] Dostál, Z. (2006a). An Optimal Algorithm for Bound and Equality Constrained Quadratic Programming Problems with Bounded Spectrum. *Computing*, 78 :311–328.
- [Dostál, 2006b] Dostál, Z. (2006b). Inexact Semimonotonic Augmented Lagrangians with Optimal Feasibility Convergence for Convex Bound and Equality Constrained Quadratic Programming. *SIAM Journal on Numerical Analysis*, 43(1) :96–115.
- [Dostál, 2009] Dostál, Z. (2009). *Optimal quadratic programming algorithms : with applications to variational inequalities*. Springer, New York, springer optim. appl. edition.
- [Dostál et al., 2003] Dostál, Z., Friedlander, A., and Santos, S. A. (2003). Augmented Lagrangians with adaptive precision control for quadratic programming with equality constraints. *SIAM J. Opt.*, 13(4) :1120–1140.
- [Dostál and Horák, 2003] Dostál, Z. and Horák, D. (2003). Scalability and FETI based algorithm for large discretized variational inequalities. *Mathematics and Computers in Simulation*, 61(3) :347–357.
- [Dostál and Horák, 2004] Dostál, Z. and Horák, D. (2004). Scalable FETI with optimal dual penalty for a variational inequality. *Numerical linear algebra with applications*, 11(56) :455–472.
- [Dostál et al., 2006] Dostál, Z., Horák, D., and Kučera, R. (2006). Total FETI—an easier implementable variant of the FETI method for numerical solution of elliptic PDE. *International Journal for Numerical Methods in Biomedical Engineering*, 22(12) :1155–1162.
- [Dostál et al., 2005] Dostál, Z., Horák, D., Kučera, R., Vondrák, V., Haslinger, J., Dobiáš, J., and Pták, S. (2005). FETI based algorithms for contact problems : scalability, large displacements and 3D Coulomb friction. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5) :395–409.
- [Dostál et al., 2009] Dostál, Z., Horák, D., and Stefanica, D. (2009). A scalable FETI–DP algorithm with non-penetration mortar conditions on contact interface. *Journal of Computational and Applied Mathematics*, 231(2) :577–591.
- [Dostál et al., 2012] Dostál, Z., Kozubek, T., Brzobohaty, T., Markopoulos, A., and Vlach, O. (2012). Scalable TFETI with optional preconditioning by conjugate projector for transient frictionless contact problems of elasticity. *Computer Methods in Applied Mechanics and Engineering*, 247-248 :37–50.
- [Dostál et al., 2010] Dostál, Z., Kozubek, T., Vondr, V., Brzobohaty, T., and Markopoulos, A. (2010). Scalable TFETI algorithm for the solution of multibody contact problems of elasticity. *International Journal for Numerical Methods in Engineering*, 82(11) :1384–1405.

- [Dostál and Kučera, 2010] Dostál, Z. and Kučera, R. (2010). An optimal algorithm for minimization of quadratic functions with bounded spectrum subjected to separable convex inequality and linear equality constraints. *SIAM J. Opt.*, 20(6) :2913–2938.
- [Dostál and Schoberl, 2002] Dostál, Z. and Schoberl, J. (2002). Minimizing quadratic functions over non-negative cone with the rate of convergence and finite termination. *SIAM J. Opt.*, pages 1–21.
- [Dryja and Widlund, 1989] Dryja, M. and Widlund, O. B. (1989). *Towards a unified theory of domain decomposition algorithms for elliptic problems*. New York University, Courant Institute of Mathematical Sciences, Division of Computer Science.
- [Dryja and Widlund, 1990] Dryja, M. and Widlund, O. B. (1990). Iterative methods for large linear systems. In Kincaid, D. R. and Hayes, L., editors, *Proceedings of the Conference on Iterative Methods for Large Linear Systems*, chapter Some Domain Decomposition Algorithms for Elliptic Problems, pages 273–291. Academic Press Professional, Inc., San Diego, CA, USA.
- [Dryja and Widlund, 1994] Dryja, M. and Widlund, O. B. (1994). Domain decomposition algorithms with small overlap. *SIAM Journal on Scientific Computing*, 15(3) :604–620.
- [Dureisseix, 1997] Dureisseix, D. (1997). *Une Approche Multi-échelles pour des Calculs de Structures sur Ordinateurs à Architecture Parallèle*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Dureisseix and Farhat, 2001] Dureisseix, D. and Farhat, C. (2001). A numerically scalable domain decomposition method for the solution of frictionless contact problems. *International Journal for Numerical Methods in Engineering*, 50(12) :2643–2666.
- [Dureisseix et al., 2003] Dureisseix, D., Ladevèze, P., and Schrefler, B. (2003). A LATIN computational strategy for multiphysics problems : application to poroelasticity. *International Journal for Numerical Methods in Engineering*, 56 :1489–1510.
- [Duval et al., 2016] Duval, M., Passieux, J. C., Salaün, M., and Guinard, S. (2016). Non-intrusive Coupling : Recent Advances and Scalable Nonlinear Domain Decomposition. *Archives of Computational Methods in Engineering*, 23(1) :17–38.
- [Duvaut, 1971] Duvaut, G. et Lions, J. (1971). Élasticité avec frottement. *Journal de Mécanique*.
- [Duvaut, 1976] Duvaut, G. et Lions, J. (1976). *Inequalities in mechanics and physics*. Springer.
- [EDF R&D,] EDF R&D. Code-aster. www.code-aster.org.
- [Embar et al., 2010] Embar, A., Dolbow, J., and Harari, I. (2010). Imposing Dirichlet boundary conditions with Nitsche’s method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*.
- [Farhat et al., 2005] Farhat, C., Avery, P., Tezaur, R., Engineering, M., and Li, J. (2005). FETI-DPH : a dual-primal domain decomposition method for acoustic scattering. *Journal of Computational Acoustics*, 13(03) :499–524.
- [Farhat et al., 1995] Farhat, C., Chen, P.-S., and Mandel, J. (1995). A scalable lagrange multiplier based domain decomposition method for time-dependent problems. *International Journal for Numerical Methods in Engineering*, 38(22) :3831–3853.

- [Farhat et al., 1998] Farhat, C., Chen, P.-S., Mandel, J., and Roux, F. X. (1998). The two-level {FETI} method part ii : Extension to shell problems, parallel implementation and performance results. *Computer Methods in Applied Mechanics and Engineering*, 155(1–2) :153 – 179.
- [Farhat et al., 2001] Farhat, C., Lesoinne, M., Patrick, L., Pierson, K., and Rixen, D. (2001). FETI-DP : a dual–primal unified FETI method—part I : A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7) :1523–1544.
- [Farhat et al., 2000a] Farhat, C., Macedo, A., and Lesoinne, M. (2000a). A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numerische Mathematik*, 85 :283–308.
- [Farhat et al., 2000b] Farhat, C., Macedo, A., Lesoinne, M., Roux, F.-X., Magoulès, F., and Bourdonnaie, A. D. L. (2000b). Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems. *Computer Methods in Applied Mechanics and Engineering*, 184(2-4) :213–239.
- [Farhat and Mandel, 1998] Farhat, C. and Mandel, J. (1998). The two-level FETI method for static and dynamic plate problems Part I : An optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155(1-2) :129–151.
- [Farhat et al., 1994] Farhat, C., Mandel, J., and Roux, F. (1994). Optimal convergence properties of the FETI domain decomposition method. *Computer Methods in Applied Mechanics and Engineering*, 115(3-4) :365–385.
- [Farhat and Roux, 1991] Farhat, C. and Roux, F.-X. (1991). A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6) :1205–1227.
- [Fortin, 1975] Fortin, M. (1975). Minimization of some non-differentiable functionals by the augmented lagrangian method of hestenes and powell. *Applied Mathematics and Optimization*, 2(3) :236–250.
- [Friedlander and Martinez, 1994] Friedlander, A. and Martinez, J. M. (1994). On the maximization of a concave quadratic function with box constraints. *SIAM J. Opt.*, 4(1) :177–192.
- [Gander, 2006] Gander, M. J. (2006). Optimized Schwarz Methods. *SIAM Review*, 44(2) :699–731.
- [Gander et al., 2008] Gander, M. J. et al. (2008). Schwarz methods over the course of time. *Electron. Trans. Numer. Anal.*, 31(5) :228–255.
- [Gander et al., 2007] Gander, M. J., Halpern, L., and Magoules, F. (2007). An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. *International journal for numerical methods in fluids*, 55(2) :163–175.
- [Gendre et al., 2009] Gendre, L., Allix, O., and Gosselet, P. (2009). Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Computational Mechanics*, 44 :233–245.
- [Giannakopoulos, 1989] Giannakopoulos, A. (1989). The return mapping method for the integration of friction constitutive relations. *Computers & Structures*, 32(1) :157–167.

- [Glowinski and Le Tallec, 1990] Glowinski, R. and Le Tallec, P. (1990). Augmented Lagrangian Interpretation of the Nonoverlapping Schwarz Alternating Method. In Glowinski, R., Périaux, R., and Widlund, O. B., editors, *Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 224–231. SIAM, Philadelphia.
- [Gosselet and Rey, 2007] Gosselet, P. and Rey, C. (2007). Non-overlapping domain decomposition methods in structural mechanics. *Archives of computational methods in engineering*, 13(4) :515–572.
- [Gosselet et al., 2015] Gosselet, P., Rixen, D., Roux, F. X., and Spillane, N. (2015). Simultaneous FETI and block FETI : Robust domain decomposition with multiple search directions. *International Journal for Numerical Methods in Engineering*, 104(10).
- [GrabCAD,] GrabCAD. www.grabcad.com.
- [Griebel and Oswald, 1995] Griebel, M. and Oswald, P. (1995). On the abstract theory of additive and multiplicative Schwarz algorithms. *Numerische Mathematik*, 70(2) :163–180.
- [Guguin et al., 2014] Guguin, G., Allix, O., Gosselet, P., and Guinard, S. (2014). Nonintrusive coupling of 3D and 2D laminated composite models based on finite element 3D recovery. *International Journal for Numerical Methods in Engineering*, 98(5) :324–343.
- [Guguin et al., 2016] Guguin, G., Allix, O., Gosselet, P., and Guinard, S. (2016). On the computation of plate assemblies using realistic 3D joint model : a non-intrusive approach. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1) :16.
- [Hansbo, 2005] Hansbo, P. (2005). Nitsche’s method for interface problems in computational mechanics. *GAMM-Mitteilungen*, 28(2) :183–206.
- [Hansbo et al., 2004] Hansbo, P., Hermansson, J., and Svedberg, T. (2004). Nitsche’s method combined with space-time finite elements for ALE fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41 SPEC. ISS.) :4195–4206.
- [Hestenes, 1969] Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5) :303–320.
- [Hild, 2002] Hild, P. (2002). On finite element uniqueness studies for Coulomb’s frictional contact model. *Int. J. Appl. Math. Comput. Sci.*, 12(1) :41–50.
- [Johansson et al., 2016] Johansson, A., Garzon, M., and Sethian, J. A. (2016). A three-dimensional coupled Nitsche and level set method for electrohydrodynamic potential flows in moving domains. *Journal of Computational Physics*, 309 :88–111.
- [Juntunen and Stenberg, 2009] Juntunen, M. and Stenberg, R. (2009). Nitsche’s method for general boundary conditions. *Mathematics of computation*, 78(267) :1353–1374.
- [Kerfriden, 2008] Kerfriden, P. (2008). *Stratégie de décomposition de domaine à trois échelles pour la simulation du délaminage dans les stratifiés*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Kerfriden et al., 2009] Kerfriden, P., Allix, O., and Gosselet, P. (2009). A three-scale domain decomposition method for the 3D analysis of debonding in laminates. *Computational Mechanics*, 44 :343–362.

- [Kikuchi and Oden, 1988] Kikuchi, N. and Oden, J. T. (1988). *Contact problems in elasticity : a study of variational inequalities and finite element methods*. SIAM.
- [Klawonn and Rheinbach, 2010] Klawonn, A. and Rheinbach, O. (2010). Highly scalable parallel domain decomposition methods with an application to biomechanics. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 90(1) :5–32.
- [Klawonn et al., 2008] Klawonn, A., Rheinbach, O., and Widlund, O. B. (2008). Analysis of a Feti–Dp Algorithm on Irregular Subdomains in the Plane. *SIAM Journal on Numerical Analysis*, 46(5) :2484–2504.
- [Klawonn and Widlund, 2001] Klawonn, A. and Widlund, O. B. (2001). FETI and Neumann–Neumann iterative substructuring methods : connections and new results. *Communications on pure and applied Mathematics*, 54(1) :57–90.
- [Kozubek et al., 2013] Kozubek, T., Vondrák, V., Menšík, M., Horák, D., Dostál, Z., Hapla, V., Kabelíková, P., and Čermák, M. (2013). Total FETI domain decomposition method and its massively parallel implementation. *Advances in Engineering Software*, 60 :14–22.
- [Kučera, 2008] Kučera, R. (2008). Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints. *SIAM J. Opt.*, 19(2) :846–862.
- [Kudawoo et al., 2011a] Kudawoo, A. D., Lebon, F., Abbas, M., De-Soza, T., and Rosu, I. (2011a). Etude de la robustesse d’un algorithme basé sur le lagrangien stabilisé pour la résolution des problèmes de contact et de frottement. In *10e colloque national en calcul des structures*, Giens, France.
- [Kudawoo et al., 2011b] Kudawoo, D., Lebon, F., Abbas, M., De Soza, T., and Rosu, I. (2011b). Two algorithms based on Stabilized Lagrangian formulation to solve frictional contact problems. In *2nd International Conference on Computational Contact Mechanics*, Hannover, Germany.
- [Ladevèze, 1985] Ladevèze, P. (1985). Sur une famille d’algorithmes en mécanique des structures. *Comptes-rendus des séances de l’Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l’univers, sciences de la terre*, 300(2) :41–44.
- [Ladevèze and Dureisseix, 1999] Ladevèze, P. and Dureisseix, D. (1999). Une nouvelle stratégie de calcul micro/macro en mécanique des structures. *Comptes Rendus de l’Académie des Sciences - Series IIB - Mechanics-Physics-Astronomy*, 327(12) :1237–1244.
- [Ladevèze et al., 2001] Ladevèze, P., Loiseau, O., and Dureisseix, D. (2001). A micro–macro and parallel computational strategy for highly heterogeneous structures. *International Journal for Numerical Methods in Engineering*, 52(12) :121–138.
- [Ladevèze and Nouy, 2003] Ladevèze, P. and Nouy, A. (2003). On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 192(28-30) :3061–3087.
- [Ladevèze et al., 2010] Ladevèze, P., Passieux, J. C., and Néron, D. (2010). The LATIN multiscale computational method and the Proper Generalized Decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21-22) :1287–1296.

- [Ladevèze, 1999] Ladevèze, P. (1999). *Nonlinear computational structural mechanics : new approaches and non-incremental methods of calculation*. Springer-Verlag, New-York.
- [Lions, 1988] Lions, P.-L. (1988). On the Schwarz alternating method. I. In *First international symposium on domain decomposition methods for partial differential equations*, pages 1–42. Paris, France.
- [Lions, 1990] Lions, P.-L. (1990). On the Schwarz alternating method. III : a variant for nonoverlapping subdomains. In *Third international symposium on domain decomposition methods for partial differential equations*, volume 6, pages 202–223. SIAM Philadelphia, PA.
- [Loiseau, 2001] Loiseau, O. (2001). *Une stratégie de calcul multiéchelle pour les structures hétérogènes*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Lorong, 1994] Lorong, P. (1994). *Sur une approche parallèle pour le calcul des structures : comportement sur calculateurs parallèles*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Maday and Magoulès, 2007] Maday, Y. and Magoulès, F. (2007). Optimized Schwarz methods without overlap for highly heterogeneous media. *Comput. Methods Appl. Mech. Engrg.*, 196 :1541–1553.
- [Magoulès and Gbikpi-Benissan, 2016] Magoulès, F. and Gbikpi-Benissan, G. (2016). JACK : an asynchronous communication kernel library for iterative algorithms. *Journal of Supercomputing*, pages 1–20.
- [Magoulès et al., 2004a] Magoulès, F., Ivanyi, P., and Topping, B. (2004a). Non-overlapping Schwarz methods with optimized transmission conditions for the Helmholtz equation. *Comput. Methods Appl. Mech. Engrg.*, 193 :4797–4818.
- [Magoulès et al., 2004b] Magoulès, F., Roux, F.-X., and Salmon, S. (2004b). Optimal discrete transmission conditions for a nonoverlapping domain decomposition method for the Helmholtz equation. *SIAM J. SCI. COMPUT.*, 25(5) :1497–1515.
- [Magoulès et al., 2015] Magoulès, F., Szyld, D. B., and Venet, C. (2015). Asynchronous optimized schwarz methods with and without overlap. Technical report, Research Report 15-06-19, Department of Mathematics, Temple University.
- [Mahjoubi et al., 2009] Mahjoubi, N., Gravouil, A., and Combescure, A. (2009). Coupling subdomains with heterogeneous time integrators and incompatible time steps. *Computational Mechanics*, 44(6) :825–843.
- [Mandel, 1993] Mandel, J. (1993). Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3) :233–241.
- [Mandel and Brezina, 1993] Mandel, J. and Brezina, M. (1993). Balancing domain decomposition : Theory and performance in two and three dimensions. Technical report, University of Colorado at Denver, Denver, CO, USA.
- [Mandel and Dohrmann, 2003] Mandel, J. and Dohrmann, C. R. (2003). Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical linear algebra with applications*, 10(7) :639–659.
- [Mandel et al., 1987] Mandel, J., McCormick, S., and Bank, R. (1987). Variational multigrid theory. *Multigrid methods*, 3 :131–177.

- [Mandel and Sousedík, 2007] Mandel, J. and Sousedík, B. (2007). Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Computer methods in applied mechanics and engineering*, 196(8) :1389–1399.
- [Mandel et al., 2012] Mandel, J., Sousedík, B., and Šístek, J. (2012). Adaptive BDDC in three dimensions. *Mathematics and Computers in Simulation*, 82(10) :1812–1831.
- [Mandel and Tezaur, 2001] Mandel, J. and Tezaur, R. (2001). On the convergence of a dual-primal substructuring method. *Numerische Mathematik*, 88(3) :543–558.
- [Mandel et al., 1999] Mandel, J., Tezaur, R., and Farhat, C. (1999). A scalable substructuring method by lagrange multipliers for plate bending problems. *SIAM Journal on Numerical Analysis*, 36(5) :1370–1391.
- [Matsokin A. M., 1985] Matsokin A. M., N. S. V. (1985). The Schwarz alternation method in a subspace. *Soviet Math. (Iz. VUZ)*, 29 :78–84.
- [More and Toraldo, 1991] More, J. J. and Toraldo, G. (1991). On the solution of large quadratic programming problems with bound constraints. *SIAM J. Opt.*, 1(1) :93–113.
- [Nabben, 2003] Nabben, R. (2003). Comparisons between multiplicative and additive Schwarz iterations in domain decomposition methods. *Numerische Mathematik*, 95(1) :145–162.
- [Negrello et al., 2017] Negrello, C., Gosselet, P., and Rey, C. (2017). A new impedance accounting for short and long range effects in mixed substructured formulations of nonlinear problems. *submitted in International Journal for Numerical Methods in Engineering*.
- [Néron, 2007] Néron, D. (2007). *Sur une stratégie de calcul pour les problèmes multiphysiques*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Nitsche, 1970] Nitsche, J. (1970). Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen in der Mathematik an der Universität Hamburg*.
- [Nouy, 2003] Nouy, A. (2003). *Une stratégie de calcul multiéchelle avec homogénéisation en temps et en espace pour le calcul de structures fortement hétérogènes*. PhD thesis, ENS Cachan.
- [Odièvre, 2009] Odièvre, D. (2009). *Sur une stratégie de calcul en dynamique transitoire en présence de variabilité paramétrique*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Pang, 1990] Pang, J.-S. (1990). Newton’s Method for B-Differentiable Equations. *Mathematics of Operations Research*, 15(2) :311–341.
- [Parret-Fréaud et al., 2017] Parret-Fréaud, A., Rey, V., Gosselet, P., and Rey, C. (2017). Improved recovery of admissible stress in domain decomposition methods - application to heterogeneous structures and new error bounds for FETI-DP. *International Journal for Numerical Methods in Engineering*.
- [Parsons and Hall, 1990] Parsons, I. D. and Hall, J. F. (1990). The multigrid method in solid mechanics : Part I - Algorithm description and behaviour. *International Journal for Numerical Methods in Engineering*, 29(4) :719–737.
- [Passieux, 2009] Passieux, J.-c. (2009). *Approximation radiale et méthode LATIN multi-échelle en temps et en espace*. PhD thesis, Ecole Normale Supérieure de Cachan.

- [Passieux et al., 2013] Passieux, J. C., Réthoré, J., Gravouil, A., and Baietto, M. C. (2013). Local/global non-intrusive crack propagation simulation using a multigrid X-FEM solver. *Computational Mechanics*, 52(6) :1381–1393.
- [Pellegrini and Roman, 1996] Pellegrini, F. and Roman, J. (1996). SCOTCH : A Software Package for Static Mapping by Dual Recursive Bipartitioning of Process and Architecture Graphs. *High-Performance Computing and Networking*, 1067 of th :493–498.
- [Polyak, 1969] Polyak, B. T. (1969). The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics*, 9(4) :94–112.
- [Powell, 1969] Powell, M. J. D. (1969). A method for nonlinear constraints in minimization problems. In Fletcher, R., editor, *Optimization*, pages 283–298. Academic Press, New York.
- [Qi and Sun, 1993] Qi, L. and Sun, J. (1993). A nonsmooth version of Newton’s method. *Mathematical Programming*, 58(1-3) :353–367.
- [Relun et al., 2013] Relun, N., Néron, D., and Boucard, P. A. (2013). A model reduction technique based on the PGD for elastic-viscoplastic computational analysis. *Computational Mechanics*, 51(1) :83–92.
- [Renard, 2013] Renard, Y. (2013). Generalized Newton’s methods for the approximation and resolution of frictional contact problems in elasticity. *Computer Methods in Applied Mechanics and Engineering*, 256 :38–55.
- [Říha et al., 2017] Říha, L., Brzobohatý, T., and Markopoulos, A. (2017). Hybrid parallelization of the total FETI solver. *Advances in Engineering Software*, 103 :29–37.
- [Rockafellar, 1973] Rockafellar, R. T. (1973). The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and applications*, 12(6) :555–562.
- [Rotscher, 1927] Rotscher, F. (1927). *Die Maschinenelemente*. Springer-Verlag, Berlin, Germany.
- [Roulet et al., 2011] Roulet, V., Champaney, L., and Boucard, P.-a. (2011). A parallel strategy for the multiparametric analysis of structures with large contact and friction surfaces. *Advances in Engineering Software*, 42(6) :347–358.
- [Saad, 2003] Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.
- [Saad and Sosonkina, 1999] Saad, Y. and Sosonkina, M. (1999). Enhanced Parallel Multicolor Preconditioning Techniques for Linear Systems. *PPSC*.
- [Saavedra et al., 2017] Saavedra, K., Allix, O., Gosselet, P., Hinojosa, J., and Viard, A. (2017). An enhanced nonlinear multi-scale strategy for the simulation of buckling and delamination on 3D composite plates. *Computer Methods in Applied Mechanics and Engineering*, 317 :952–969.
- [Saavedra Redlich, 2012] Saavedra Redlich, K. (2012). *Stratégie multiéchelle pour l’analyse du couplage flambage-délaminage de composites stratifiés*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Schwarz, 1870] Schwarz, H. A. (1870). Ueber einen Grenzübergang durch alternirendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15 :272–286.

- [Simo and Laursen, 1992] Simo, J. C. and Laursen, T. (1992). An augmented Lagrangian treatment of contact problems involving friction. *Computers & Structures*, 42(1) :97–116.
- [Smith et al., 2004] Smith, B., Bjorstad, P., Gropp, W. D., and Gropp, W. (2004). *Domain decomposition : parallel multilevel methods for elliptic partial differential equations*. Cambridge university press.
- [Spillane et al., 2011a] Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. (2011a). A robust two-level domain decomposition preconditioner for systems of PDEs. *Comptes Rendus Mathématique*, 349(23-24) :1255–1259.
- [Spillane et al., 2011b] Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. (2011b). A robust two-level domain decomposition preconditioner for systems of PDEs. *Comptes Rendus Mathématique*, 349(23-24) :1255–1259.
- [Spillane and Rixen, 2013] Spillane, N. and Rixen, D. J. (2013). Automatic spectral coarse spaces for robust finite element tearing and interconnecting and balanced domain decomposition algorithms. *International Journal for Numerical Methods in Engineering*.
- [Tallec et al., 1991] Tallec, P., Roeck, Y., and Vidrascu, M. (1991). Domain decomposition methods for large linearly elliptic three-dimensional problems. *Journal of Computational and Applied Mathematics*, 34(1) :93 – 117.
- [Vauchez, 1991] Vauchez, P. (1991). *Simulation numérique des processus de mise en forme par la méthode à grand increment de temps*. PhD thesis, Université Pierre et Marie Curie.
- [Violeau, 2007] Violeau, D. (2007). *Stratégie de calcul pour l'analyse à l'échelle "micro" des endommagements jusqu'à rupture des composites stratifiés*. PhD thesis, Ecole Normale Supérieure de Cachan.
- [Vitse, 2017] Vitse, M. (2017). *Model-order reduction for the parametric analysis of damage in reinforced concrete structures*. PhD thesis, Université Paris-Sacaly / Ecole Normale Supérieure de Cachan.
- [Říha, 2015] Říha, Brzobohatý, M. (2015). Highly Scalable FETI Methods in ESPRESO. In P. Iványi, B. T., editor, *Proceedings of the Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*.
- [Wriggers and Laursen, 2006] Wriggers, P. and Laursen, T. A. (2006). *Computational contact mechanics*, volume 30167. Springer.
- [Wriggers and Zavarise, 2008] Wriggers, P. and Zavarise, G. (2008). A formulation for frictionless contact problems using a weak form introduced by nitsche. *Computational Mechanics*, 41(3) :407–420.
- [Yastrebov, 2011] Yastrebov, V. (2011). *Computational contact mechanics : geometry, detection and numerical techniques*. PhD thesis, École Nationale Supérieure des Mines de Paris.

Titre : Une méthode de décomposition de domaine mixte non-intrusive pour le calcul parallèle d'assemblages

Mots clés : Décomposition de domaine, Calcul non-intrusif, Assemblages, Contact

Résumé : Les assemblages sont des éléments critiques pour les structures industrielles. De fortes non-linéarités de type contact frottant, ainsi que des précharges mal maîtrisées rendent complexe tout dimensionnement précis. Présents en très grand nombre sur les structures industrielles (quelques millions pour un A380), cela implique de raffiner les modèles localement et donc de gérer des problèmes numériques de très grandes tailles. Les nombreuses interfaces de contact frottant sont des sources de difficultés de convergence pour les simulations numériques. Il est donc nécessaire de faire appel à des méthodes robustes. Il s'agit d'utiliser des méthodes itératives de décomposition de domaine, permettant de gérer des modèles numériques extrêmement grands, couplées à des techniques adaptées afin de prendre en compte les non-linéarités de contact aux interfaces entre sous-domaines. Ces méthodes de décomposition de domaine restent encore très peu utilisées dans un cadre industriel. Des développements internes aux codes éléments finis sont souvent nécessaires et freinent ce transfert du monde académique au monde industriel.

Nous proposons, dans ces travaux de thèse, une mise-en-oeuvre non intrusive de ces méthodes de décomposition de domaine : c'est-à-dire sans développement au sein du code source. En particulier, nous nous intéressons à la méthode Latin dont la philosophie est particulièrement adaptée aux problèmes non linéaires. La structure est décomposée en sous-domaines reliés

entre eux au travers d'interfaces. Avec la méthode Latin, les non-linéarités sont résolues séparément des aspects linéaires. La résolution est basée sur un schéma itératif à deux directions de recherche qui font dialoguer les problèmes linéaires globaux et les problèmes locaux non linéaires.

Au cours de ces années de thèse, nous avons développé un outil totalement non intrusif sous *Code_Aster* permettant de résoudre par une technique de décomposition de domaine mixte des problèmes d'assemblage. Les difficultés posées par le caractère mixte de la méthode Latin sont résolues par l'introduction d'une direction de recherche non locale. Des conditions de Robin sur les interfaces des sous-domaines sont alors prises en compte simplement sans modifier les sources de *Code_Aster*. Nous avons proposé une réécriture algébrique de l'approche multi-échelle assurant l'extensibilité de la méthode. Nous nous sommes aussi intéressés à coupler la méthode Latin en décomposition de domaine à un algorithme de Krylov. Appliqué uniquement à un problème sous-structuré avec interfaces parfaites, ce couplage permet d'accélérer la convergence. Des structures préchargées avec de nombreuses interfaces de contact frottant ont été traitées. Des simulations qui n'auraient pu être menées par un calcul direct sous *Code_Aster* ont été réalisées via cette stratégie de décomposition de domaine non intrusive.

Title : A non-invasive mixed domain decomposition for the parallel computation of assemblies

Keywords : Domain decomposition, Non-invasive computation, Assembly, Contact

Abstract : Assemblies are critical elements for industrial structures. Strong non-linearities such as frictional contact, as well as poorly controlled preloads make complex all accurate sizing. Present in large numbers on industrial structures (a few million for an A380), this involves managing numerical problems of very large size. The numerous interfaces of frictional contact are sources of difficulties of convergence for the numerical simulations. It is therefore necessary to use robust but also reliable methods. The use of iterative methods based on domain decomposition allows to manage extremely large numerical models. This needs to be coupled with adapted techniques in order to take into account the nonlinearities of contact at the interfaces between subdomains. These methods of domain decomposition are still scarcely used in industries. Internal developments in finite element codes are often necessary, and thus restrain this transfer from the academic world to the industrial world.

In this thesis, we propose a non-intrusive implementation of these methods of domain decomposition : that is, without development within the source code. In particular, we are interested in the Latin method whose philosophy is particularly adapted to nonlinear problems. It consists in decomposing

the structure into sub-domains that are connected through interfaces. With the Latin method the non-linearities are solved separately from the linear differential aspects. Then the resolution is based on an iterative scheme with two search directions that make the global linear problems and the nonlinear local problems dialogue.

During this thesis, a totally non-intrusive tool was developed in *Code_Aster* to solve assembly problems by a mixed domain decomposition technique. The difficulties posed by the mixed aspect of the Latin method are solved by the introduction of a non-local search direction. Robin conditions on the subdomain interfaces are taken into account simply without modifying the sources of *Code_Aster*. We proposed an algebraic rewriting of the multi-scale approach ensuring the extensibility of the method. We were also interested in coupling the Latin method in domain decomposition to a Krylov algorithm. Applied only to a substructured problem with perfect interfaces, this coupling accelerates the convergence. Preloaded structures with numerous contact interfaces have been processed. Simulations that could not be carried out by a direct computation with *Code_Aster* were performed via this non-intrusive domain decomposition strategy.

